PATENTS
ODS-9

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

PATENT APPLICATION

Applicants   : Connie T. Marshall et al..

Application No. : 09/609,073   Confirmation No. : 2964

Filed   : June 30, 2000

For   : INTERACTIVE WAGERING SYSTEMS AND
                   METHODS FOR RECORDING WAGERING EVENTS

Group Art Unit   : 3714

Examiner   : Steven L. Ashburn

Mail Stop Appeal Briefs - Patents
Hon. Commissioner for Patents
P.O. Box 1450
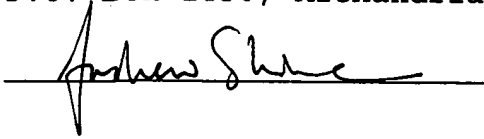Alexandria, Virginia 22313-1450

EXPRESS MAIL CERTIFICATION

Express Mail Label No. EV626910645US

Date of Deposit: March 11, 2005

I hereby certify that this certification and the following papers and fees:

1. Appeal Brief;
2. Authorization to Charge Deposit Account (in duplicate);
3. Check in the amount of $2,160.00 (extension fee); and
4. Check in the amount of $500.00 (filing fee)

03/16/2005 EABUBAK1 00000045 09609073
02 FC:1255   2160.00 OP

are being deposited with the United States Postal Service
"Express Mail Post Office to Addressee" service under 37 C.F.R.
§ 1.10 on the date indicated above and are addressed to Mail
Stop Appeal Briefs - Patents Hon. Commissioner for Patents,
P.O. Box 1450, Alexandria, VA 22313-1450.

Name: _Andrew Shiue_

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants   : Connie T. Marshall et al.

Application No.   : 09/609,073   Confirmation No. : 2964

Filed   : June 30, 2000

For   : INTERACTIVE WAGERING SYSTEMS AND METHODS
                       FOR RECORDING WAGERING EVENTS

Group Art Unit   : 3714

Examiner   : Steven L. Ashburn

Mail Stop Appeal Briefs - Patents
Hon. Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

AUTHORIZATION TO CHARGE DEPOSIT ACCOUNT

Sir:

The Director is hereby authorized to charge any additional fee due, or credit any overpayment, in connection with the accompanying Appeal Brief submitted concurrently herewith, to Deposit Account No. 06-1075. A duplicate copy of this Authorization is submitted herewith.

Respectfully submitted,

Adam M. Saltzman
Registration No. 52,188
Agent for Applicants
Fish & Neave IP Group
Ropes & Gray LLP
Customer No. 1473
1251 Avenue of the Americas
New York, NY 10020-1105
Tel.: (212) 596-9000
Fax : (212) 596-9090

Express Mail No.
EV626910645US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants   :   Connie T. Marshall et al.

Application No.   :   09/609,073   Confirmation No. : 2964

Filed   :   June 30, 2000

For   :   INTERACTIVE WAGERING SYSTEMS AND METHODS
                      FOR RECORDING WAGERING EVENTS

Group Art Unit   :   3714

Examiner   :   Steven L. Ashburn

Mail Stop Appeal Briefs - Patents
Hon. Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF

Sir:

      Applicants are filing this Appeal Brief in support of their appeal from the final rejection of claims 2-19 and 38-48[*] in the Office Action dated February 11, 2004. A Notice of Appeal for this case was filed on August 11, 2004.

---

[*]   Claims 1 and 37 were also rejected in the final Office Action dated February 11, 2004. However, in applicants' Reply to Final Office Action dated July 12, 2004, applicants cancelled claims 1 and 37. For purposes of Appeal, these amendments were entered in the Advisory Action dated August 12, 2004. Therefore, applicants will refer to the rejection in the final Office Action dated February 11, 2004 as being of claims 2-19 and 38-48 throughout this Appeal Brief.

Express Mail No.
EV626910645US

Applicants hereby petition for a five-month extension of time for submitting this Appeal Brief. With the extension, the Appeal Brief is due on or before March 11, 2005. A check in the amount of $2160.00, in payment of the extension fee required under 37 C.F.R. §§ 1.136(a) and 1.17(a)(5), is enclosed herewith. A check in the amount of $500.00, in payment of the filing fee required under 37 C.F.R. § 41.20(b)(2), is also enclosed. The Director is hereby authorized to charge any additional fees that may be due in connection with this Appeal Brief, or credit any overpayment of the same, to Deposit Account No. 06-1075. A separate Authorization to Charge Deposit Account is enclosed for that purpose (in duplicate).

I.   Introduction

In the final Office Action dated February 11, 2004, the Examiner finally rejected claims 2-19 and 38-48 under 35 U.S.C. § 103(a) as being unpatentable over Brenner et al., U.S. Patent No. 5,830,068 ("Brenner '068") in view of Dan Wagner et al., The Human Factors Design Guide ("HFDG"), and Lawler et al., U.S. Patent No. 5,805,763 ("Lawler"). Claims 2-19 and 38-48 have been rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-59 of Brenner et al., U.S. Patent No. 6,004,211 (hereinafter "Brenner '211").

In view of the arguments and authorities set forth below, the Board should find these rejections to be in error and should reverse the Examiner.

II. Appendices

This Brief has the following appendices:

Claims Appendix

Appendix A: Copy of claims 2-19 and 38-48 involved in this appeal;

Evidence Appendices

Appendix B: Copy of the Office Action dated July 21, 2003;

Appendix C: Copy of the Reply to Office Action dated November 20, 2003;

Appendix D: Copy of the Final Office Action dated February 11, 2004;

Appendix E: Copy of the Reply to Final Office Action dated July 12, 2004;

Appendix F: Copy of the Advisory Action dated August 12, 2004;

Appendix G: Copy of Brenner et al., U.S. Patent No. 5,830,068 (entered in the record by the Examiner in the Office Action dated August 27, 2001 and initialed by the Examiner in the Information Disclosure Statement on August 17, 2001);

Appendix H: Copy of Lawler et al., U.S. Patent No. 5,805,763 (entered in the record by the Examiner in the Office Action dated March 21, 2002 and initialed by the Examiner in the Information Disclosure Statement on August 17, 2001);

Appendix I: Copy of introductory section and
              section 8 of Dan Wagner et al.,
              "The Human Factors Design Guide"
              (pages 8-1 to 8-157 entered in the
              record by the Examiner in the
              Office Action dated July 21,
              2003); and

Appendix J: Copy of Brenner et al., U.S.
              Patent No. 6,004,211 (entered in
              the record by the Examiner in the
              Office Action dated March 21,
              2002).

III. Identification of Real Party in Interest

          Applicants respectfully advise the Board that the
real party in interest in the above-identified patent
application is ODS Properties, Inc., a corporation
organized and existing under the laws of the State of
Delaware, and having an office and place of business at
6701 Center Drive West, Los Angeles, CA 90045, which is
the assignee of this application.

IV.   Related Appeals and Interferences

          Applicants respectfully advise the Board that
there are no other appeals or interferences known to
applicants, their legal representative, or their assignee
that will directly affect or be directly affected by or
have a bearing on the Board's decision in the pending
appeal.

4

V.   Status of Claims

        Claims 2-19 and 38-48 are rejected in this
application and are on appeal. Claims 1, 20-37 and 49-56
have been cancelled.

VI.   Status of Amendments

        Applicants' Reply to Final Office Action dated
July 12, 2004 cancelled independent claims 1 and 37 without
prejudice. Claims 2-18 and 38-47 were previously dependent
on claims 1 and 37, respectively. Pursuant to 37 C.F.R.
§ 1.116, claims 2, 4, 11, 14-18, 38 and 43-46 were amended
such that claims 2-18 and 38-47 would be dependent on
independent claims 19 and 48, respectively. For purposes
of Appeal, these amendments were entered in the Advisory
Action dated August 12, 2004.

VII.   Summary of Claimed Subject Matter

        Applicants' invention, as defined by independent
claims 19 and 48, generally relates to systems and methods
for allowing users to wager on and record wagering events
(see, e.g., applicants' specification, page 29, line 31 to
page 30, line 9 and page 35, line 31 to page 36, line 16).
A user is allowed to create and place a wager for a given
race (see, e.g., applicants' specification, page 38, line
18 to page 41, line 26; see also, e.g., applicants'
drawings, FIGS. 8-14). The user is automatically provided
with an opportunity to record the given race in response to
the user placing the wager for a given race (see, e.g.,
applicants' specification, page 41, lines 18-26; see also,
e.g., applicants' drawings, FIG. 14). The given race is
recorded (see, e.g., applicants' specification, page 41,

5

line 27 to page 42, line 11 and page 43, line 30 to page
44, line 3; see also, e.g., applicants' drawings, FIG. 15).

VIII. Grounds of Rejection to be Reviewed on Appeal

     The following grounds of rejection are to be
reviewed on this appeal:

     (a) claims 2-19 and 38-48 stand rejected under 35
U.S.C. § 103(a) as being unpatentable over Brenner '068 in
view of HFDG and Lawler; and

     (b) claims 2-19 and 38-48 stand rejected under
the judicially created doctrine of obviousness-type double
patenting as being unpatentable over claims 1-59 of
Brenner '211.

IX.   Argument

    A.   Rejection of Claims 2-19 and 38-48
        Under 35 U.S.C. § 103(a)

     In the final Office Action dated February 11,
2004, the Examiner rejected claims 2-19 and 38-48 under 35
U.S.C. § 103(a) as being obvious over Brenner '068 in view
of HFDG and Lawler. Applicants respectfully traverse this
rejection and request that it be overturned for at least
the reasons set forth below.

          i.   Brenner, Lawler And HFDG Fail To Show Or
               Suggest Automatically Providing A User
               With An Opportunity To Record A Race In
               Response To Placing A Wager on That Race

     As stated in the final Office Action dated
February 11, 2004, Brenner '068 discloses an interactive
wagering system that allows users to create and place
wagers by interacting with an interactive wager-creation
interface (see Brenner '068, e.g., FIGS. 36-39 and 41-44).
Brenner '068 also shows that a selectable option associated

6

with a given race may be presented, and that the user may
select the option to record the race (see Brenner '068,
column 28, lines 4-23 and FIG. 49). However, the Examiner
concedes that Brenner '068 does not show automatically
providing the user with an opportunity to record the given
race in response to the user placing the wager for the
given race (see February 11, 2004 Office Action, page 5).
The Examiner attempts to show this feature of applicants'
claims using the disclosure from Lawler.

Lawler generally relates to an interactive
television program guide for allowing users to browse and
select television programs (see Lawler, abstract). The
Examiner contends that Lawler automatically provides users
with an opportunity to record a given event in response to
the user placing an order for a given item in FIGS. 4A, 4B
and 6-10 (see February 11, 2004 Office Action, page 5).
Applicants respectfully disagree. As illustrated in
FIG. 6, program options menu 136 includes both order
button 138 and record button 130. Applicants submit that
these two buttons provide distinct options for a user, and
that in response to the selection of order button 138, the
option to record a program is not automatically provided to
the user, as contended by the Examiner. The flow diagram
in FIGS. 4A and 7 of Lawler further illustrates this point.

FIG. 4A shows the process by which order
button 138 and record button 130 are displayed. More
specifically, at block 200, the program time guide of
FIG. 3 is displayed. The program time guide includes
program schedule information of various programs (see
Lawler, column 7, lines 29-32 and column 9, lines 36-38).
If the user selects a future program in the program time
guide, block 238 indicates that the system displays program

7

options menu 136 (illustrated in FIG. 6) along with the
program time guide (previously displayed in FIG. 3) (see
Lawler, column 10, line 65 - column 11, line 4). As
mentioned above, program options menu 136 of FIG. 6
includes, *inter alia*, the distinct options of order
button 138 and record button 130.

FIG. 7 shows the process that occurs if one of
the buttons in program options menu 136 is selected and
demonstrates that these buttons provide distinct options,
thus undermining the Examiner's contention that Lawler
shows automatically providing a user with an opportunity to
record a given event in response to placing an order for a
given item. In particular, if order button 138 in program
options menu 136 is selected, the system proceeds to
block 306 where a menu to facilitate ordering is then
displayed at block 308. At block 310, the system monitors
and implements the user's selections from the ordering menu
and then at block 312 the system returns to the program
time guide of FIG. 3 (see Lawler, column 11, lines 36-44).
Nowhere in this process is it shown or suggested that the
user is automatically provided with an opportunity to
record a program, as the Examiner contends. In fact, once
the system returns to the program time guide of FIG. 3, the
process, as outlined in the flow diagram of FIGS. 4A and 7,
returns to block 200. Following the steps described above
beginning at block 200, in order to record the program, the
user would have to select the future program again from the
program time guide. At that point, program options
menu 136 is displayed again and only then can the user
select record button 130 to record the program. Therefore,
Lawler does not show or suggest automatically providing
users with an opportunity to record a program in response

8

to the user placing an order for the program. Thus, since Lawler does not show or suggest automatically providing users with an opportunity to record a program in response to the user placing an order for the program, it cannot show or suggest "automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race," as required in applicants' independent claims 19 and 48.

The February 11, 2004 Office Action appears to rely on HFDG to provide the motivation to combine Brenner '068 with Lawler, but it also relies on three sections of HFDG to modify the user interface disclosed in Brenner '068 to show features of applicants' claims. However, HFDG is described as "a comprehensive reference tool that will help human factors professionals within the Federal Aviation Administration (FAA) and contractor organizations to efficiently carry out FAA human factors policy" (HFDG, page i of the Foreword). In addition, HFDG "provides reference information to assist in the selection, analysis, design, development, and evaluation of new and modified FAA systems, facilities, and equipment" (HFDG, page 1-1). Although HFDG does, as the Examiner suggests, refer to various fundamental goals for implementing human-computer interfaces, HFDG fails to refer to designing interactive wagering interfaces (see February 11, 2004 Office Action, page 6).

Nevertheless, the Examiner first refers to section 8.1.6.3 of HFDG which states that "[t]he system or application shall provide the user whatever information is required to guide control entries" (HFDG, page 8-14; see also February 11, 2004 Office Action, page 6). As an example, section 8.1.6.3 states that "[p]rompts may be

incorporated into a display at any point in a transaction sequence <u>that will be helpful</u>" (HFDG, page 8-14, emphasis added). Applicants respectfully submit that this section merely shows, in a broad sense, that it would be helpful to provide prompts. However, nowhere is it shown or suggested in this section that it would be helpful to automatically provide the user with an opportunity to record a race in response to a user placing a wager for the race.

The Examiner also points to sections 8.1.11.1.7 and 8.1.11.3.5 of HFDG, which refer to providing menu options, in an attempt to show features of applicants' claims (see February 11, 2004 Office Action, page 6). More specifically, section 8.1.11.1.7 states that menus should display all options that are available to a user at a step in a transaction sequence (see HFDG, page 8-20). Section 8.1.11.3.5 states that critical or frequently selected options should be easily accessible to a user (see HFDG, page 8-22). The Examiner contends that HFDG suggests modifying Brenner '068 to display all available options and to give access to frequently used functions (see February 11, 2004 Office Action, page 6). However, applicants submit that the Examiner has not shown or suggested that providing the user with an opportunity to record a race in response to placing a wager is an available option in the transaction sequence. Furthermore, applicants submit that the Examiner has not shown or suggested that applicants' claimed feature is critical or a frequently used function. In fact, applicants' claims require providing the user with the opportunity to record a race in response to the user placing a wager for the race. If recording a race is considered critical or frequently used, then it would seem

10

that this feature should be part of the wager creation process, and not displayed in response to placing a wager.

Therefore, at least because Brenner '068, Lawler and HFDG do not show or suggest "automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race," applicants respectfully submit that the Board should reverse the obviousness rejection of independent claims 19 and 48 under 35 U.S.C. § 103(a).

ii. The Examiner Failed To Provide Sufficient Motivation To Combine

The Examiner has failed to provide sufficient motivation for combining the references to justify the assertion of a § 103 rejection. See In re Rouffet, 149 F.3d 1350, 1355 (Fed. Cir. 1998) ("When a rejection depends on a combination of prior art references, there must be some teaching, suggestion, or motivation to combine the references"); see also MPEP § 2142 and 2143.01. It is well-settled that an Examiner can "satisfy this burden only by showing some objective teaching ... that would lead [one of ordinary skill in the art] to combine the relevant teachings of the references." In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988).

As mentioned above, the Examiner conceded that Brenner '068 does not automatically provide the user with an opportunity to record the given race in response to the user placing the wager for the given race. The Examiner attempted to the modify Brenner '068 to include this feature by relying on broad teaching in Lawler of automatically providing users with an opportunity to record a given event in response to the user placing an order for an item (see February 11, 2004 Office Action, page 6).

11

However, as demonstrated above in subsection (i), Lawler does not show or suggest this teaching. Therefore, Lawler lacks the requisite motivation to modify Brenner '068 to show or suggest with objective evidence applicants' claimed feature.

As further demonstrated above, the sections of HFDG cited by the Examiner failed to show or suggest automatically providing the user with an opportunity to record a race in response to a user placing a wager for the race. In fact, some of the other sections of HFDG, which were not cited by the Examiner, when taken as a whole, teach away from applicants' claimed feature. For example, section 8.1.14.2.2 of HFDG states that only information that is relevant to a task should be included in a system or application (see HFDG, page 8-32). However, nowhere is it shown or suggested that providing the user with the opportunity to record a race is relevant to the task of placing a wager on the race. Therefore, HFDG would seem to suggest to not provide the user with the opportunity to record a race in response to the user placing a wager on the race.

Another example is found in section 8.1.14.4.2 of HFDG which states that a user's effort should be minimized by reducing the number of keystrokes required of users (see HFDG, page 8-33). Applicants' claims require providing the user with the opportunity to record the race in response to placing a wager. This may add keystrokes, however, which is discouraged by HFDG. If HFDG suggests minimizing keystrokes, then HFDG would seem to suggest that the user should be provided with the opportunity to record the race while placing a wager, not in response to placing the wager.

These two exemplary teachings, illustrate how HFDG teaches away from applicants' claimed approaches. In addition, applicants respectfully submit that there is no objective evidence of record, other than applicants' disclosure, that would lead one skilled in the art to modify Brenner '068, Lawler and HFDG to automatically provide users with an opportunity to record a race "in response to the user placing the wager for the given race" as specified by applicants' claims. Without objective evidence of a motivation to modify the references to arrive at applicants' claimed approach, the Examiner "simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability," a practice that is insufficient as a matter of law. See In re Dembiczak, 175 F.3d 994, 999 (Fed. Cir. 1999); see also In re Lee, 277 F.3d 1338, 1344 (Fed. Cir. 2002) ("[i]t is improper, in determining whether a person of ordinary skill would have been led to a combination of references, simply to use that which the inventor taught against its teacher").

Therefore, because neither Brenner '068, Lawler, HFDG, nor their combination show or suggest applicants' claimed approach, applicants respectfully submit that a *prima facie* case of obviousness had not been met and that the Board should reverse the obviousness rejection of independent claims 19 and 48 under 35 U.S.C. § 103(a). Because claims 19 and 48 are allowable, clams 2-18 and 38-47, which depend from claims 19 and 48, respectively, are also allowable.

13

B.   Double Patenting Rejection of
     Claims 2-19 and 38-48

          Claims 2-19 and 38-48 were rejected under the
judicially created doctrine of obviousness-type double
patenting (analogous to a rejection under 35 U.S.C. § 103
according to MPEP § 804(II)(B)(1)) as being unpatentable
over claims 1-59 of Brenner '211. Applicants respectfully
submit, however, that the obviousness-type double patenting
rejection is improper.

          It is well settled that, in cases where double
patenting may be at issue, "it must always be carefully
observed that the ... patent [used as the basis for a
double patenting rejection] is not 'prior art' under either
section 102 or section 103 of the 1952 Patent Act (35
U.S.C. as amended)." In re Boylan, 392 F.2d 1017, 1018
(C.C.P.A. 1968); see also In re Braithwaite, 379 F.2d 594,
600, n.4 (C.C.P.A. 1967) ("While analogous to the non-
obviousness requirement of 35 U.S.C. § 103, that section is
not itself involved in double patenting rejections because
the patent principally underlying the rejection is not
prior art"). Indeed, the courts have determined that a
double patenting rejection is reserved for situations
"where patents are not citable as a reference against each
other and therefore can not be examined for compliance with
the rule that only one patent is available per invention."
Eli Lilly & Co. v. Barr Labs., 251 F.3d 955, 966 (Fed.
Cir. 2001) (Circuit Judge Newman dissenting, in a separate
opinion, on the Court's refusal to reconsider the case en
banc); see also General Foods Corp. v. Studiengesellschaft
Kohle mbH, 972 F.2d 1272, 1278 (Fed. Cir. 1992).

          With respect to the present case, Brenner '068
was filed on September 8, 1995 and issued on November 3,

1998, which is before the date of applicants' claimed invention.* Accordingly, the claims and disclosure of Brenner '068 are statutory prior art under 35 U.S.C. § 102(a). The Examiner agreed with the applicants' arguments in the November 20, 2003 Reply to Office Action that the double patenting rejection based on the claims of Brenner '068 was improper and withdrew the rejection (see February 11, 2004 Office Action, page 2).

The Examiner, however, maintained the double patenting rejection with respect to Brenner '211, which was filed on August 24, 1998, issued on December 21, 1999 and claims priority from Brenner '068. The Examiner contended that "obviousness-type double patenting is determined based on comparison of claims, not disclosures. . . the fact that two inventions share a common disclosure is no matter in determining obviousness-type double patenting" (Office Action, page 4). However, Brenner '211 claims priority from Brenner '068 and the subject matter of the claims in Brenner '211 is fully supported and disclosed in Brenner '068. Therefore, applicants respectfully submit that the obviousness-type double patenting rejection based on the subject matter of claims 1-59 of Brenner '211 is improper because this subject matter is disclosed in Brenner '068 which is available as prior art under 35 U.S.C. § 103. Accordingly, the Board should reverse the obviousness-type double patenting rejection.

However, even if the double patenting rejection is found to be proper, applicants respectfully submit that,

---

* Applicants' non-provisional patent application was filed on January 30, 2000 and claims priority from U.S. provisional patent application No. 60/142,174, filed July 1, 1999.

15
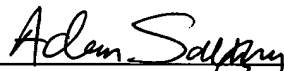
for the same reasons set forth above in connection with the rejection under § 103 over Brenner '068, applicants' claims 2-19 and 38-48 are not obvious in view of claims 1-59 of Brenner '211.

X.   Conclusion

For the reasons set forth above, applicants respectfully submit that claims 2-19 and 38-48 are in condition for allowance. The Examiner's rejections of these claims should be reversed.

Respectfully submitted,

Adam M. Saltzman
Registration No. 52,188
Agent for Applicants
Fish & Neave IP Group
Ropes & Gray LLP
Customer No. 1473
1251 Avenue of the Americas
New York, NY 10020-1105
Tel.: (212) 596-9000
Fax : (212) 596-9090

CLAIMS APPENDIX A
CLAIMS ON APPEAL

2.   The method defined in claim 19 further comprising allowing the user to select yes or no in response to the option to record the given race.

3.   The method defined in claim 2 wherein the given race is recorded in response to the user selecting yes to the option to record the given race.

4.   The method defined in claim 19 wherein the given race is recorded in a personal archive.

5.   The method defined in claim 4 wherein the personal archive is located at the user equipment.

6.   The method defined in claim 4 wherein the personal archive is located remote from the user equipment.

7.   The method defined in claim 4 further comprising using the interactive wagering application to allow the user to access the personal archive and view previously recorded races.

8.   The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding track name.

9. The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding race number.

10. The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding date.

11. The method defined in claim 19 wherein the user equipment is user television equipment.

12. The method defined in claim 11 wherein the given race is recorded with a videocassette recorder.

13. The method defined in claim 11 wherein the given race is recorded with a digital video recorder.

14. The method defined in claim 19 wherein the user equipment is user computer equipment.

15. The method defined in claim 19 wherein the user equipment is user telephone equipment.

16. The method defined in claim 19 wherein the given race is recorded in real-time.

17. The method defined in claim 19 wherein the given race is recorded after the race has taken place.

18. The method defined in claim 19 wherein the user is charged a fee for recording the given race.

19. A method for a user at user equipment to interactively wager on races with an interactive wagering application implemented using the user equipment, comprising:

allowing the user to create and place a wager for a given race;

automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race; and

recording the given race.

38. The interactive wagering system defined in claim 48 wherein the user equipment is user television equipment.

39. The interactive wagering system defined in claim 38 wherein the control circuitry is located within a set-top box.

40. The interactive wagering system defined in claim 38 wherein the display device is a television.

41. The interactive wagering system defined in claim 38 wherein the recording device is a videocassette recorder.

42. The interactive wagering system defined in claim 38 wherein the recording device is a digital video recorder.

43. The interactive wagering system defined in claim 48 wherein the user equipment is user computer equipment.

44. The interactive wagering system defined in claim 48 wherein the user equipment is user telephone equipment.

45. The interactive wagering system defined in claim 48 wherein the recording device is located remote from the user equipment.

46. The interactive wagering system defined in claim 48 wherein the given race is recorded in a personal archive.

47. The interactive wagering system defined in claim 46 wherein the control circuitry is further configured to allow the user to access the recording device and view previously recorded races.

48. An interactive wagering system in which an interactive wagering application is implemented on user equipment that provides a user with an opportunity to place wagers on races to be run, comprising:

control circuitry configured to allow the user to create and place a wager for a given race, wherein the user is automatically provided with an opportunity to record the given race in response to the user placing the wager for the given race; and

a recording device that records the given race.

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/609,073	06/30/2000	CONNIE T MARSHALL	ODS-9	2964

1473   7590   07/21/2003

FISH & NEAVE
1251 AVENUE OF THE AMERICAS
50TH FLOOR
NEW YORK, NY 10020-1105

RECEIVED

JUL 2 4 2003

FISH & NEAVE PATENT DEPT.
REFERRED TO
NOTED BY

EXAMINER
ASHBURN, STEVEN L

ART UNIT	PAPER NUMBER
3714	

DATE MAILED: 07/21/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

REMINDER - PLEASE INITIAL
THIS DOCUMENT TO INDICATE
THAT YOU HAVE SEEN IT

DOCKETED FOR Oct 21, 2003

PTO-90C (Rev. 07-01)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *11 June 2003* .

2a) ☐ This action is FINAL.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-19 and 37-48* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-19 and 37-48* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.

    If approved, corrected drawings are required in reply to this Office action.

12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

        1. ☐ Certified copies of the priority documents have been received.

        2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

        3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

    a) ☐ The translation of the foreign language provisional application has been received.

15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

Notice of References Cited

	Application/Control No.	Applicant(s)/Patent Under Reexamination
Notice of References Cited	09/609,073	MARSHALL ET AL.
	Examiner	Art Unit	Page 1 of 1
	Steven Ashburn	3714	

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Dan Wagner et al., 'The Human Factors Design Guide', DOT/FAA/CT-96/1 (Jan. 15, 1996), pp. 8-1 to 8-157
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

DETAILED ACTION

Withdraw of Finality

In view of the applicant's arguments filed on May 9, 2003 (paper no. 15) and the subsequent telephone interview on June 3, 2003 (paper no. 17), PROSECUTION IS HEREBY REOPENED. A new grounds rejection is set forth below.

Double Patenting

Claims 1-19 and 37-48 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over either claims 1-59 of Brenner et al., U.S. Patent 6,004,211 (Dec. 21, 1999) (hereinafter "*Brenner '211*") or claims 1-132 of Brenner et al., U.S. Patent 5,830,068 (Nov. 3, 1998) (hereinafter "*Brenner 068*") each Dan Wagner et al., 'The Human Factors Design Guide', DOT/FAA/CT-96/1 (Jan. 15, 1996) (hereinafter "*HFDG*") and Lawler et al., U.S. 5,805,763 (Sep. 8, 1998)..

Brenner '068 and *Brenner '211* disclose "off-track" wagering systems having an interactive user interface allowing users to review racing information and place bets. *See abstract.* The through the interface, users may select a desired racetrack and race and view odds, pools, and payoff amounts for a variety of wager types. *See id.* To place a wager, the user selects a wager type, wager amount, and the desired runners. *See id.* Account information can be reviewed and the user can transfer funds from a bank account to an account used for wagering. *See id.* Racing videos can be viewed while the user reviews odds and places bets. *See id.* Alternatively, video clips of past races can be ordered. *See id.* Related advertisements can be presented using text or video clips. *See id.* Merchandise may be ordered interactively. *See id.* Finally, information regarding system usage may be gathered. *See id.*

In regards to claims 1, 19, 37 and 48: *Brenner '068* and *Brenner '211* teach the following features:

a.   Allowing a user to create and place a wager for a given race by interacting with a plurality of wager creation options. *See '068, 1, 2, 33, 35, 48-57, See '211, claims 1, 19, 37, 58 and 59*

b.   Providing the user with an option to record the given race while the user is interacting with a plurality of wager creation options. *See '068, claims 69, 89, 92, 93, 96, 100, 125. See '211, claims 1, 19, 37, 58 and 59.*

c.   Recording the given race on a video recording device. *See '068, claims 94, 125. See '211, claims 2, 5, 36, 37 and 53.*

Brenner '068 teaches all the features of the claims except the particular combination of automatically prompting the user to decide whether to record the race while interacting the plurality of wager creation options. Regardless of the deficiency, this feature would have been obvious to an artisan in view of the *HFDG* and *Lawler*.

The *HFDG* provides reference information to assist in the selection, analysis, design, development, and evaluation of new and modified Federal Aviation Administration (FAA) systems and equipment. *See abstract.* A preliminary edition was a draft standard developed at the Human Factors Laboratory of the FAA Technical Center. *See id.* This 1996 edition converts the preliminary draft document to a guide and incorporates expert comments that were collected in 1994 and 1995 from selected reviewers. *See id.* It is primarily focused on FAA ground systems and equipment such as those that are managed and maintained by Airway Facilities. *See id.* This guide covers a broad range of human factors topics that pertain to automation, maintenance, human interfaces, workplace design,

documentation, system security, safety, the environment, and anthropometry. *See id.* The document also includes extensive human-computer interface guidance.

Section 8 of the *HFDG* is directed particularly to human-computer interfaces. *See §8.1.* This section contains criteria and guidelines governing human-computer interfaces. The topics covered include: modes of human-computer interaction, basic screen design, windowing, data entry, data display, user guidance, data communication, input devices, and accommodating people with disabilities. *See id.* In regards to the claimed invention, the HFDG teaches fundamental concepts of interactive user interfaces that were within the knowledge of an artisan at the time of the invention. Portions of the section which are relevant to the applicant's claims are provided below:

8.1 USER-COMPUTER INTERACTION

8.1.1 General

8.1.1.1 Consistent control actions.
Interactive control actions should be consistent in form, means, and consequence from one transaction to another, from one task to another, and from one application to another. **Discussion.** This guideline is extremely important for users of multiple applications. For example, if a user of a system being designed or selected must control several diverse operating systems or inconsistent control functions, then high error rates, extensive training, and low human reliability may be a consequence.

8.1.1.2 System matched to user abilities.
Interactive control systems should be adaptable to individual differences and should accommodate the variety of user abilities expected, whether novice or expert. If applicable, systems and applications should provide relatively helpful or self-explanatory operations for novice or infrequent users, and relatively efficient operations for experienced users.

8.1.1.6 Simplicity.
Interactive control shall be simple, flexible, and adaptive, as well as consistent and compatible with the lowest anticipated user skill level. Interactive control shall be logical in terms of user task sequences and functions.

8.1.1.7 Minimal user actions.
Interactive control logic should permit completion of a task with the minimum number of actions, consistent with user abilities.

8.1.1.12 Minimal memory load.
The short-term memory requirements on users should be minimized by such means as making displays and interactive sequences self-evident and by providing on-line help and tutorials.

8.1.1.24 Prompts.
If a user must perform several actions to complete a task, the application should prompt the user with the actions that need to be performed.

8.1.6 Transaction and control options

8.1.6.3 Prompting control entries.
The system or application shall provide the user whatever information is required to guide control entries. **Examples.** Prompts may be incorporated into a display at any point in a transaction sequence that will be helpful, or prompts may appear in response to a request for help. The selected prompts must be used consistently.

8.1.6.4 List of basic control options.
A list of basic control options that are always available to a user shall be easily displayable. This list can serve as a "home base" or starting point for control entries. An example is the system-level menu.

8.1.6.7 Option presentation.
The options presented in a list of basic options should be grouped, labeled, and ordered according to their: (1) logical function, (2) sequence, (3) frequency, or (4) criticality of use. If these ordering schemes are in conflict, default to the higher level order.

8.1.8 Interaction method

8.1.8.1 Selection of interaction type.
The interaction type shall be selected to be appropriate to the task requirements, the characteristics of the system, and the abilities of the users. The appropriateness of the major types of interaction for these requirements, characteristics, and abilities are listed here and summarized in exhibit 8.1.8.1.

8.1.8.3 Hierarchical levels.
If hierarchical levels are used to control a process or sequence, the number of levels shall be minimized. Display and input formats shall be similar within levels, and the system shall indicate the current position within a sequence (see also paragraph 8.1.11.3.4).

8.1.11 Menus and menu selection
The use of menus as an interaction method is widespread, often in conjunction with other methods, direct manipulation, in particular. Menus are usable with little or no training on the part of the user. If the meanings of the options are clear, the user can be guided step-by-step through an application. Menus do have some disadvantages, however; they can slow down an experienced user; they can occupy a considerable amount of display space; and, in complex sequences, users may become lost in the menu structure.

8.1.11.1.6 Number of options.
The number of options in a menu should not be more than ten or less than three (same as paragraph 8.3.7.3.6).

8.1.11.1.7 Display of all options.
A menu shall display explicitly and completely all options available to a user at the current step in a transaction sequence.

8.1.11.1.11 Shortcuts for experienced users.
Experienced users should have a way to bypass the menu structure for frequently accessed options (see also paragraph 8.1.11.3.13).

8.1.11.3 Hierarchical menus
Large or complex menus can be presented as hierarchical menus. **Definition. A hierarchical menu is a** large menu that is organized as a multi-level, branching structure of smaller menus in which an option in a higher level menu is the name of another menu at the next lower level. The options in the lowest level menus are such things as commands or values; they are not the names of other menus.

8.1.11.3.1 When to use.
Hierarchical menus should be used if there are many options (more than 10), and the options can be organized in a branching structure.

8.1.11.3.4 Minimum number of levels.
A hierarchical menu structure should minimize the number of selections required to reach the desired option. This implies the use of broad, shallow structures as opposed to narrow, deep ones.

8.1.11.3.5 Easy selection of important options.
Hierarchical menus should permit immediate user access to critical or frequently selected options.

8.1.11.3.13 Bypassing menu selections. The system or application should allow a user to bypass a series of menu selections by making an equivalent command entry (see also paragraph 8.1.11.1.11).

8.1.14 Query and natural language
This section contains criteria and guidelines for database queries.
Definitions. A data base is a set of interrelated data stored in a computer. A query is the process of specifying, locating, and retrieving data matching specified characteristics from a data base.

8.1.14.2 Query screen design

8.1.14.2.1 Applicable criteria and guidelines.
Query screen design shall conform to the criteria and guidelines in sections 8.3, 8.4, and 8.5.

8.1.14.2.2 Relevant information only.
Query screens should include only information that is relevant to the task, that is, information necessary to perform actions, make decisions, or answer questions.

8.1.14.2.3 Frequently-used information.
The most frequently used information should be located in the upper left portion of a screen and, if multiple screens are involved, on the first screen or screens.

8.1.14.4.2 Minimal user effort.
The number of keystrokes required of users should be minimized.

8.1.15 Graphical controls
Icons may be used to represent operations, processes, and data structures graphically, and they may be used as a means of exercising control over system functions, components, and data structures.

8.2 BASIC SCREEN DESIGN AND OPERATION
Screen design refers to the way information is arranged and presented on a display screen. Different systems and applications can perform a great variety of tasks. Some systems rely heavily on data bases and do not require immediate user response to information displayed on their screens. Other systems, such as control systems, require that the users make immediate decisions and issue commands based on information displayed to them. The designer needs to understand the primary function of the system being developed to provide an effective screen design.

8.2.1 Principles, features, and functions

8.2.1.1 General principles

8.2.1.1.1 Simplicity.
Information should be presented simply and in a well-organized manner. Ways to achieve simplicity include the following:
a. The screen should appear to be orderly and clutter-free.
b. Information should be presented in consistent, predictable locations.
c. The language used should be plain and simple.
d. The means for moving around the screen and to related screens should be simple.
e. Interrelationships should be indicated clearly.

8.2.1.1.2 Logical grouping.
Data items on a screen should be grouped on the basis of some logical principle.

8.2.1.1.3 Minimal movement. Screens should be designed to minimize both eye movement and pointer movement.

8.2.1.1.4 What information to display.
The information to be displayed should be prioritized so that the most important or critical information can be displayed all the time, and less important or critical information can be displayed upon a user's request.

8.2.1.8.1 Minimizing the user's short-term memory load.
Windows should be designed to minimize the short-term memory load placed on a user as he or she performs the task called for by the window. A window should contain all relevant information and should allow a user to complete the task without having to refer to additional information.

8.2.1.9.2 Matching window layout to users' "Natural" patterns. Window layout should conform to users' natural scanning order and probable selection sequences. Usually, the order will be from left to right and top to bottom. For example, in button sets and menus, the most frequent choice should appear in the leftmost or top position.

8.2.1.9.3 Minimal user effort.
The amount of pointer movement and the number of keystrokes required to complete a task should be minimized.

8.3 WINDOWING
Windows provide a convenient and easy to use means of organizing many of the interactive aspects of a system or application and presenting them to a user.
Definition. A window is a rectangular area on the screen that provides a visual means for interaction with an application. Applications also use windows to provide information to the user. This section contains criteria and guidelines for window components, appearance, and states, for window controls and operations, for menus and text in windows, and for a variety of special purpose windows.
Caveat. Much of the material contained in section 8.6 may be very closely tied to a particular scheme or model for implementing windows and handling window management operations. The scheme being alluded to in any one rule may not be the only way of handling windows, nor is it the only recommended, approved, or acceptable way of doing so. To imply otherwise might violate the intent (if not the letter) of paragraph 4.1.10 of this standard. The authors of this guideline have, to the extent possible, removed guidelines that would have eliminated or restricted a particular window management system. For example, the OSF/Motif, Open Look, Apple Macintosh, and Microsoft Windows window management systems all offer similar, but slightly differing models for accomplishing many of the same windowing functions. To prematurely focus upon and exclusively adopt any single one of these management systems would do a disservice to the users of this proposed guideline. However, to simply strike out all such implementation-specific referential paragraphs within this section would result in removing a great deal of potentially helpful or useful design guidance information. The editors of these guidelines have chosen to retain these paragraphs for the potential value they might offer as examples of at least one acceptable method of implementing a windows operating environment.

8.3.3 Window controls
This section contains criteria and guidelines for window controls. **Definitions.** A control is any object that allows a user to perform an action. Controls include buttons, menu options, settings, sliders, text fields, and check boxes. A **push button** is a control that appears as a bounded area (for example, a rectangle or oval) on a window.

8.3.12.2 Data entry windows

8.3.12.2.1 Data entry window elements.

A data entry window should contain: (1) a title that describes the purpose or contents of the window, (2) a set of labeled fields, (3) vertical or horizontal scroll bars or both, if the contents do not fit in the window's working area, and (4) controls appropriate to the task.

Definition. A **data entry window** is a window that contains a set of labeled fields for entering, changing, and deleting data. It may also contain labeled data display fields, which a user cannot change.

8.3.12.2.2 Data window organization. The organization of a data entry window should be consistent with the task it represents. For example, data fields should be arranged by sequence of use, frequency of use, or importance, with related fields grouped together and separated from unrelated fields. If users will enter data from hard copy forms, the data entry field organization should correspond to that of the hard copy.

8.3.12.2.3 Multipage data entry windows.
If the contents of a set of data entry fields do not fit the window's working area, a. the window should provide users the ability to page, scroll, or both through the entire set, and b. if the fields are arranged in rows, columns, or both, the labels of the rows or columns should remain in place when the rows or columns scroll or page.

8.3.12.2.4 Push buttons in data entry windows.
If a data entry window contains push buttons, the buttons should be placed in a row at the bottom of the working area, visually separated from the data fields.

8.3.12.2.5 Controls for data entry windows.
A data entry window should contain the controls appropriate to the task.

8.4 DATA ENTRY
Data entry refers to user actions involving the input of data into a computer system, and the system's response to the user actions. The data entry methods covered in this section are: (1) selection from menus, (2) form filling, (3) direct manipulation, and (4) the keyboard entry of text. Additional topics covered in this section include the entry of tabular and graphic data, and the validation of entered data.

8.4.1 Menus
Menus are often useful in data entry, for example, to list files that may be retrieved, or to list the acceptable entries for a field in a form. Menus of this sort are often too long to display in their entirety. In that case, a portion of the menu is displayed and a scrolling capability is provided.

8.4.1.2 Hierarchical menus

8.4.1.2.1 When to use.
Hierarchical menus should be used if the number of options is more than ten and the options can be organized into a meaningful hierarchy. Note. A hierarchical structure may be more cumbersome and keystroke intensive than a longer, single-level structure. Thus, if a long list of options is obviously and logically organized, it will be easier to use than a hierarchical structure. For instance, consider a list of type sizes numerically ordered or a long list font alternatives logically organized.

8.4.1.2.2 Applicable rules.
If hierarchical menus are used for data entry, they shall conform to the rules in section 8.1.11.3.

8.4.1.4 Pop-up menus
Pop-up menus can be very useful in data entry. They can present to a user the permissible entries for a field, thus (1) eliminating the need for the user to remember the entries, (2) preventing invalid entries, and (3) eliminating potential typing errors. **Definition.** A **pop-up menu** is a menu that is associated with a particular object on a display, for example, a pop-up menu listing acceptable command options close to the immediate work area. This is particularly useful for large displays, where the work site may be relatively removed for the menu bar.

As provided above, the *HFDG* describes fundamental goals for implementing effective human-computer interfaces. In light of these goals, more specific prior art is discussed below.

Lawler discloses an analogous system having an interactive user interface allowing users to search and automatically record broadcast events. Similar to *Brenner*, wherein users collect race program information and place orders for racing videos, *Lawler* allows users to collect entertainment program information and place orders for event videos. *See fig. 3; col. 1:45-2:40.* The system provides on-demand delivery of event videos ranging from brief clips to full length motion pictures. *See col. 4:23-26.* Users may order videos of past, present or future events. *See fig. 7.* In specific regards to the claims, *Lawler* describes automatically prompting the user to decide whether to record an event while interacting the plurality of program selection menus. *See fig. 4(a)(b), 6-10.* As a result, the system allows users to more quickly and easily identify and select events using an interactive user-interface and to designate the selected event for recording. *See col. 1:45-50.*

In view of the *HFDG* and *Lawler*, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the off-track wagering system disclosed by *Brenner* to add the feature of automatically prompting the user to decide whether to record the race while interacting the plurality of wager creation options. *Brenner* discloses an interactive human-computer interface for off-track betting wherein a menu allows users the option of record racing videos. *Lawler* describes automatically prompting a user to decide whether to record an event while interacting with a plurality of program selection menus and thereby more easily designate a selected event for recording. *See col. 1:45-50. See fig. 4(a)(b), 6-10.* As suggested by the *HFDG*, modifying *Brenner* the add the feature would enhance the system's human-computer interface by (i) permitting completion of the task with the minimum number of actions; (ii) incorporating prompts into the display at points in a transaction sequence that will be helpful; (iii) including with the list of basic control option a selection that is always available to a user; (iv) presenting frequently used and/or critical option within the group of basic options; (v) displaying all

options available to a user at the current step in a transaction sequence; (vi) providing experienced users a way to bypass the menu structure to execute the frequently accessed option; (vii) permitting immediate access to a critical and/or frequently selected option; (viii) prioritizing the display so that an important or critical information is displayed all the time; (ix) minimizing the amount of pointer movement and the number of keystrokes required for the user to complete the task; and (x) adapting the user interface to accommodate a variety of user abilities such that it offers relatively efficient operations for experienced users. Consequently the claims are unpatentable because they are obvious when the prior art is taken as a whole by an artisan at a time prior to the invention.

In further regards to the claim 2: *Lawler* additionally describes allowing the user to selecting "order" or "cancel" in response to the option to select the given race. *See fig. 10; col. 10:60-64.* This is equivalent to the claimed feature of selecting "yes" or "no". Although the terminology is different, the method is the same.

In further regards to the claim 3: *Lawler* additionally suggests recording the race in response to the user selecting "yes" or "no". *See id.*

In further regards to the claims 4 and 46: *Lawler* additionally describes storing the event in a personal archive. *See col. 13:26-37.*

In further regards to the claim 5: *Lawler* additionally describes locating the personal event archive on the user's equipment. *See col. 2:14-23.*

In further regards to dependent claim 6 and 45: *Brenner '068* and *Brenner '211* additionally describe storing race video at a remote location. *See '068, claims 6, 32, 42. See '211, claims 2, 7.* Alternatively, *Lawler* additionally describes locating the personal archive remotely from the user's equipment. *See col. 2:24-35, 13:26-37.*

In further regards to dependent claims 7 and 47: *Brenner '068* and *Brenner '211* additionally describe allowing users to access and view stored race recordings on a display device. *See '068, 24-29. See '211, claims 4, 24.* Alternatively, *Lawler* additionally describes allowing users to access and view stored race recordings on a display device. *See col. 2:24-35, 13:26-37.*

In further regards to dependent claims 8-10: *Lawler* additionally describes allowing the user to search for videos based on search criteria including time, date, length, subject. *See 7:10-18.*

In further regards to dependent claims 11, 38 and 40: *Brenner '068* and *Brenner '211* additionally describe using television equipment as user equipment. *See '068, 16, 35, 45, 78. See '211, claim 14, 29, 46, 49.* Alternatively, *Lawler* additionally describes using television equipment as user equipment. *See fig. 1.*

In regards to claims 12 and 41: *Brenner '068* and *Brenner '211* additionally describe recording a given race with a video cassette recorder. *See '068, claims 94, 125. See '211, claims 2, 5, 36, 37 and 53.* Alternatively, *Lawler* describes recording a given event with a video cassette recorder. *See col. 1:33-42, 3:36-39.*

In regards to claim 13 and 42: *Lawler* additionally describes Recording the given race on a videocassette recorder or, alternatively, other digital recording device. *See col. 3:28-67.*

In further regards to dependent claims 14 and 43: *Brenner '068* and *Brenner '211* additionally describe employing a computer as user equipment. Notably, the examiner interprets to claims use of "terminal" to include computers. *See '068, claims 1, 2, 36. See '211, claims 14, 37, 58, 59.* Alternatively, *Lawler* additionally describes employing a computer as user equipment. *See fig. 2.*

In further regards to the claims 15 and 44: *Lawler* additionally describes employing telephone equipment as user equipment. *See col. 5:29-36.*

In further regards to dependent claim 16: *Brenner '068* and *Brenner '211* additionally describe recording the race in real-time. *See '068, claims 96, 125. See '211, claims 1, 4.* Alternatively, *Lawler* additionally describes recording the events in real-time. *See fig. 7.*

In further regards to dependent claim 17 and 47: *Brenner '068* and *Brenner '211* additionally describes recording the race after it has taken place. *See '068, claims 96, 125. See '211, claims 1, 4.* Alternatively, *Lawler* additionally describes recording the race after it has taken place. *See fig. 7.*

In further regards to dependent claim 18: *Brenner '068* and *Brenner '211* additionally describes charging a fee for recording a given race. *See '068, claims 75, 102. See '211, claims 12, 37, 46.*

In further regards to the claim 39: *Lawler* additionally describes locating control circuitry in a "set-top box". *See fig. 1.*

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness

rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-19 and 37-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brenner,

U.S. 5,830,068 (Nov. 3, 1998) in view of Dan Wagner et al., 'The Human Factors Design Guide',

DOT/FAA/CT-96/1 (Jan. 15, 1996) (hereinafter "*HFDG*") and Lawler et al., U.S. 5,805,763 (Sep. 8,

1998).

Brenner discloses an "off-track" wagering system having an interactive user interface allowing

users to review racing information and place bets. *See abstract.* The through the interface, users may

select a desired racetrack and race and view odds, pools, and payoff amounts for a variety of wager types.

See id. To place a wager, the user selects a wager type, wager amount, and the desired runners. *See id.*

Account information can be reviewed and the user can transfer funds from a bank account to an account

used for wagering. *See id.* Racing videos can be viewed while the user reviews odds and places bets.

See id. Alternatively, video clips of past races can be ordered. *See id.* Related advertisements can be

presented using text or video clips. *See id.* Merchandise may be ordered interactively. *See id.* Finally,

information regarding system usage may be gathered. *See id.*

In regards to claims 1, 19, 37 and 48: *Brenner* teaches the following features:

d.   Allowing a user to create and place a wager for a given race by interacting with a

plurality of wager creation options.

e.   Providing the user with an option to record the given race while the user is

interacting with a plurality of wager creation options. *See fig. 34(596)*

f.   Recording the given race on a video recording device. *See id.*

Brenner '068 teaches all the features of the claims except the particular combination of automatically

prompting the user to decide whether to record the race while interacting the plurality of wager creation

options. Regardless of the deficiency, this feature would have been obvious to an artisan in view of the

HFDG and *Lawler*.

The *HFDG* provides reference information to assist in the selection, analysis, design,

development, and evaluation of new and modified Federal Aviation Administration (FAA) systems and

equipment. *See abstract.* A preliminary edition was a draft standard developed at the Human Factors

Laboratory of the FAA Technical Center. *See id.* This 1996 edition converts the preliminary draft

document to a guide and incorporates expert comments that were collected in 1994 and 1995 from

selected reviewers. *See id.* It is primarily focused on FAA ground systems and equipment such as those

that are managed and maintained by Airway Facilities. *See id.* This guide covers a broad range of human

factors topics that pertain to automation, maintenance, human interfaces, workplace design,

documentation, system security, safety, the environment, and anthropometry. *See id.* The document also

includes extensive human-computer interface guidance.

Section 8 of the *HFDG* is directed particularly to human-computer interfaces. *See §8.1.* This

section contains criteria and guidelines governing human-computer interfaces. The topics covered

include: modes of human-computer interaction, basic screen design, windowing, data entry, data display,

user guidance, data communication, input devices, and accommodating people with disabilities. *See id.*

In regards to the claimed invention, the HFDG teaches fundamental concepts of interactive user interfaces that were within the knowledge of an artisan at the time of the invention. Portions of the section which are relevant to the applicant's claims are provided below:

8.1 USER-COMPUTER INTERACTION

8.1.1 General

8.1.1.1 Consistent control actions.
Interactive control actions should be consistent in form, means, and consequence from one transaction to another, from one task to another, and from one application to another. **Discussion.** This guideline is extremely important for users of multiple applications. For example, if a user of a system being designed or selected must control several diverse operating systems or inconsistent control functions, then high error rates, extensive training, and low human reliability may be a consequence.

8.1.1.2 System matched to user abilities.
Interactive control systems should be adaptable to individual differences and should accommodate the variety of user abilities expected, whether novice or expert. If applicable, systems and applications should provide relatively helpful or self-explanatory operations for novice or infrequent users, and relatively efficient operations for experienced users.

8.1.1.6 Simplicity.
Interactive control shall be simple, flexible, and adaptive, as well as consistent and compatible with the lowest anticipated user skill level. Interactive control shall be logical in terms of user task sequences and functions.

8.1.1.7 Minimal user actions.
Interactive control logic should permit completion of a task with the minimum number of actions, consistent with user abilities.

8.1.1.12 Minimal memory load.
The short-term memory requirements on users should be minimized by such means as making displays and interactive sequences self-evident and by providing on-line help and tutorials.

8.1.1.24 Prompts.
If a user must perform several actions to complete a task, the application should prompt the user with the actions that need to be performed.

8.1.6 Transaction and control options

8.1.6.3 Prompting control entries.
The system or application shall provide the user whatever information is required to guide control entries. **Examples.** Prompts may be incorporated into a display at any point in a transaction sequence that will be helpful, or prompts may appear in response to a request for help. The selected prompts must be used consistently.

8.1.6.4 List of basic control options.
A list of basic control options that are always available to a user shall be easily displayable. This list can serve as a "home base" or starting point for control entries. An example is the system-level menu.

8.1.6.7 Option presentation.
The options presented in a list of basic options should be grouped, labeled, and ordered according to their: (1) logical function, (2) sequence, (3) frequency, or (4) criticality of use. If these ordering schemes are in conflict, default to the higher level order.

8.1.8 Interaction method

8.1.8.1 Selection of interaction type.
The interaction type shall be selected to be appropriate to the task requirements, the characteristics of the system, and the abilities of the users. The appropriateness of the major types of interaction for these requirements, characteristics, and abilities are listed here and summarized in exhibit 8.1.8.1.

8.1.8.3 Hierarchical levels.
If hierarchical levels are used to control a process or sequence, the number of levels shall be minimized. Display and input formats shall be similar within levels, and the system shall indicate the current position within a sequence (see also paragraph 8.1.11.3.4).

8.1.11 Menus and menu selection
The use of menus as an interaction method is widespread, often in conjunction with other methods, direct manipulation, in particular. Menus are usable with little or no training on the part of the user. If the meanings of the options are clear, the user can be guided step-by-step through an application. Menus do have some disadvantages, however; they can slow down an experienced user; they can occupy a considerable amount of display space; and, in complex sequences, users may become lost in the menu structure.

8.1.11.1.6 Number of options.
The number of options in a menu should not be more than ten or less than three (same as paragraph 8.3.7.3.6).

8.1.11.1.7 Display of all options.
A menu shall display explicitly and completely all options available to a user at the current step in a transaction sequence.

8.1.11.1.11 Shortcuts for experienced users.
Experienced users should have a way to bypass the menu structure for frequently accessed options (see also paragraph 8.1.11.3.13).

8.1.11.3 Hierarchical menus
Large or complex menus can be presented as hierarchical menus. **Definition.** A **hierarchical menu** is a large menu that is organized as a multi-level, branching structure of smaller menus in which an option in a higher level menu is the name of another menu at the next lower level. The options in the lowest level menus are such things as commands or values; they are not the names of other menus.

8.1.11.3.1 When to use.
Hierarchical menus should be used if there are many options (more than 10), and the options can be organized in a branching structure.

8.1.11.3.4 Minimum number of levels.
A hierarchical menu structure should minimize the number of selections required to reach the desired option. This implies the use of broad, shallow structures as opposed to narrow, deep ones.

8.1.11.3.5 Easy selection of important options.
Hierarchical menus should permit immediate user access to critical or frequently selected options.

8.1.11.3.13 Bypassing menu selections. The system or application should allow a user to bypass a series of menu selections by making an equivalent command entry (see also paragraph 8.1.11.1.11).

8.1.14 Query and natural language
This section contains criteria and guidelines for database queries.
Definitions. A **data base** is a set of interrelated data stored in a computer. A **query** is the process of specifying, locating, and retrieving data matching specified characteristics from a data base.

8.1.14.2 Query screen design

8.1.14.2.1 Applicable criteria and guidelines.
Query screen design shall conform to the criteria and guidelines in sections 8.3, 8.4, and 8.5.

8.1.14.2.2 Relevant information only.
Query screens should include only information that is relevant to the task, that is, information necessary to perform actions, make decisions, or answer questions.

8.1.14.2.3 Frequently-used information.
The most frequently used information should be located in the upper left portion of a screen and, if multiple screens are involved, on the first screen or screens.

8.1.14.4.2 Minimal user effort.
The number of keystrokes required of users should be minimized.

8.1.15 Graphical controls
Icons may be used to represent operations, processes, and data structures graphically, and they may be used as a means of exercising control over system functions, components, and data structures.

8.2 BASIC SCREEN DESIGN AND OPERATION
Screen design refers to the way information is arranged and presented on a display screen. Different systems and applications can perform a great variety of tasks. Some systems rely heavily on data bases and do not require immediate user response to information displayed on their screens. Other systems, such as control systems, require that the users make immediate decisions and issue commands based on information displayed to them. The designer needs to understand the primary function of the system being developed to provide an effective screen design.

8.2.1 Principles, features, and functions

8.2.1.1 General principles

8.2.1.1.1 Simplicity.
Information should be presented simply and in a well-organized manner. Ways to achieve simplicity include the following:
a. The screen should appear to be orderly and clutter-free.
b. Information should be presented in consistent, predictable locations.
c. The language used should be plain and simple.
d. The means for moving around the screen and to related screens should be simple.
e. Interrelationships should be indicated clearly.

8.2.1.1.2 Logical grouping.
Data items on a screen should be grouped on the basis of some logical principle.

8.2.1.1.3 Minimal movement. Screens should be designed to minimize both eye movement and pointer movement.

8.2.1.1.4 What information to display.
The information to be displayed should be prioritized so that the most important or critical information can be displayed all the time, and less important or critical information can be displayed upon a user's request.

8.2.1.8.1 Minimizing the user's short-term memory load.
Windows should be designed to minimize the short-term memory load placed on a user as he or she performs the task called for by the window. A window should contain all relevant information and should allow a user to complete the task without having to refer to additional information.

8.2.1.9.2 Matching window layout to users' "Natural" patterns. Window layout should conform to users' natural scanning order and probable selection sequences. Usually, the order will be from left to right and top to bottom. For example, in button sets and menus, the most frequent choice should appear in the leftmost or top position.

8.2.1.9.3 Minimal user effort.
The amount of pointer movement and the number of keystrokes required to complete a task should be minimized.

8.3 WINDOWING

Windows provide a convenient and easy to use means of organizing many of the interactive aspects of a system or application and presenting them to a user.

Definition. A **window** is a rectangular area on the screen that provides a visual means for interaction with an application. Applications also use windows to provide information to the user. This section contains criteria and guidelines for window components, appearance, and states, for window controls and operations, for menus and text in windows, and for a variety of special purpose windows.

Caveat. Much of the material contained in section 8.6 may be very closely tied to a particular scheme or model for implementing windows and handling window management operations. The scheme being alluded to in any one rule may not be the only way of handling windows, nor is it the only recommended, approved, or acceptable way of doing so. To imply otherwise might violate the intent (if not the letter) of paragraph 4.1.10 of this standard. The authors of this guideline have, to the extent possible, removed guidelines that would have eliminated or restricted a particular window management system. For example, the OSF/Motif, Open Look, Apple Macintosh, and Microsoft Windows window management systems all offer similar, but slightly differing models for accomplishing many of the same windowing functions. To prematurely focus upon and exclusively adopt any single one of these management systems would do a disservice to the users of this proposed guideline. However, to simply strike out all such implementation-specific referential paragraphs within this section would result in removing a great deal of potentially helpful or useful design guidance information. The editors of these guidelines have chosen to retain these paragraphs for the potential value they might offer as examples of at least one acceptable method of implementing a windows operating environment.

8.3.3 Window controls

This section contains criteria and guidelines for window controls. **Definitions.** A control is any object that allows a user to perform an action. Controls include buttons, menu options, settings, sliders, text fields, and check boxes. A **push button** is a control that appears as a bounded area (for example, a rectangle or oval) on a window.

8.3.12.2 Data entry windows

8.3.12.2.1 Data entry window elements.
A data entry window should contain: (1) a title that describes the purpose or contents of the window, (2) a set of labeled fields, (3) vertical or horizontal scroll bars or both, if the contents do not fit in the window's working area, and (4) controls appropriate to the task.

Definition. A **data entry window** is a window that contains a set of labeled fields for entering, changing, and deleting data. It may also contain labeled data display fields, which a user cannot change.

8.3.12.2.2 Data window organization. The organization of a data entry window should be consistent with the task it represents. For example, data fields should be arranged by sequence of use, frequency of use, or importance, with related fields grouped together and separated from unrelated fields. If users will enter data from hard copy forms, the data entry field organization should correspond to that of the hard copy.

8.3.12.2.3 Multipage data entry windows.
If the contents of a set of data entry fields do not fit the window's working area, a. the window should provide users the ability to page, scroll, or both through the entire set, and b. if the fields are arranged in rows, columns, or both, the labels of the rows or columns should remain in place when the rows or columns scroll or page.

8.3.12.2.4 Push buttons in data entry windows.
If a data entry window contains push buttons, the buttons should be placed in a row at the bottom of the working area, visually separated from the data fields.

8.3.12.2.5 Controls for data entry windows.
A data entry window should contain the controls appropriate to the task.

8.4 DATA ENTRY
Data entry refers to user actions involving the input of data into a computer system, and the system's response to the user actions. The data entry methods covered in this section are: (1) selection from menus, (2) form filling, (3) direct manipulation, and (4) the keyboard entry of text. Additional topics covered in this section include the entry of tabular and graphic data, and the validation of entered data.

8.4.1 Menus
Menus are often useful in data entry, for example, to list files that may be retrieved, or to list the acceptable entries for a field in a form. Menus of this sort are often too long to display in their entirety. In that case, a portion of the menu is displayed and a scrolling capability is provided.

8.4.1.2 Hierarchical menus

8.4.1.2.1 When to use.
Hierarchical menus should be used if the number of options is more than ten and the options can be organized into a meaningful hierarchy. Note. A hierarchical structure may be more cumbersome and keystroke intensive than a longer, single-level structure. Thus, if a long list of options is obviously and logically organized, it will be easier to use than a hierarchical structure. For instance, consider a list of type sizes numerically ordered or a long list font alternatives logically organized.

8.4.1.2.2 Applicable rules.
If hierarchical menus are used for data entry, they shall conform to the rules in section 8.1.11.3.

8.4.1.4 Pop-up menus
Pop-up menus can be very useful in data entry. They can present to a user the permissible entries for a field, thus (1) eliminating the need for the user to remember the entries, (2) preventing invalid entries, and (3) eliminating potential typing errors. **Definition. A pop-up menu** is a menu that is associated with a particular object on a display, for example, a pop-up menu listing acceptable command options close to the immediate work area. This is particularly useful for large displays, where the work site may be relatively removed for the menu bar.

As provided above, the *HFDG* describes fundamental goals for implementing effective human-computer

interfaces. In light of these goals, more specific prior art is discussed below.

Lawler discloses an analogous system having an interactive user interface allowing users to

search and automatically record broadcast events. Similar to *Brenner*, wherein users collect race program

information and place orders for racing videos, *Lawler* allows users to collect entertainment program

information and place orders for event videos. *See fig. 3; col. 1:45-2:40.* The system provides on-

demand delivery of event videos ranging from brief clips to full length motion pictures. *See col. 4:23-26.*

Users may order videos of past, present or future events. *See fig. 7.* In specific regards to the claims,

Lawler describes automatically prompting the user to decide whether to record an event while interacting the plurality of program selection menus. *See fig. 4(a)(b), 6-10.* As a result, the system allows users to more quickly and easily identify and select events using an interactive user-interface and to designate the selected event for recording. *See col. 1:45-50.*

In view of the *HFDG* and *Lawler*, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the off-track wagering system disclosed by *Brenner* to add the feature of automatically prompting the user to decide whether to record the race while interacting the plurality of wager creation options. *Brenner* discloses an interactive human-computer interface for off-track betting wherein a menu allows users the option of record racing videos. *Lawler* describes automatically prompting a user to decide whether to record an event while interacting with a plurality of program selection menus and thereby more easily designate a selected event for recording. *See col. 1:45-50.. See fig. 4(a)(b), 6-10.* As suggested by the *HFDG*, modifying *Brenner* the add the feature would enhance the system's human-computer interface by (i) permitting completion of the task with the minimum number of actions; (ii) incorporating prompts into the display at points in a transaction sequence that will be helpful; (iii) including with the list of basic control option a selection that is always available to a user; (iv) presenting frequently used and/or critical option within the group of basic options; (v) displaying all options available to a user at the current step in a transaction sequence; (vi) providing experienced users a way to bypass the menu structure to execute the frequently accessed option; (vii) permitting immediate access to a critical and/or frequently selected option; (viii) prioritizing the display so that an important or critical information is displayed all the time; (ix) minimizing the amount of pointer movement and the number of keystrokes required for the user to complete the task; and (x) adapting the user interface to accommodate a variety of user abilities such that it offers relatively efficient operations for experienced users. Consequently the claims are unpatentable because they are obvious when the prior art is taken as a whole by an artisan at a time prior to the invention.

In further regards to the claim 2: *Lawler* additionally describes allowing the user to selecting "order" or "cancel" in response to the option to select the given race. *See fig. 10; col. 10:60-64.* This is equivalent to the claimed feature of selecting "yes" or "no". Although the terminology is different, the method is the same.

In further regards to the claim 3: *Lawler* additionally suggests recording the race in response to the user selecting "yes" or "no". *See id.*

In further regards to the claims 4 and 46: *Lawler* additionally describes storing the event in a personal archive. *See col. 13:26-37.*

In further regards to the claim 5: *Lawler* additionally describes locating the personal event archive on the user's equipment. *See col. 2:14-23.*

In further regards to dependent claim 6 and 45: *Brenner '068* additionally describes storing race video at a remote location. *See 7:4-20, 27:23-39.*. Alternatively, *Lawler* additionally describes locating the personal archive remotely from the user's equipment. *See col. 2:24-35, 13:26-37.*

In further regards to dependent claims 7 and 47: *Brenner '068* additionally describes allowing users to access and view stored race recordings on a display device. *See fig. 1; 27:60-64.* Alternatively, *Lawler* additionally describes allowing users to access and view stored race recordings on a display device. *See col. 2:24-35, 13:26-37.*

In further regards to dependent claims 8-10: *Lawler* additionally describes allowing the user to search for videos based on search criteria including time, date, length, subject. *See 7:10-18.*

In further regards to dependent claims 11, 38 and 40: *Brenner '068* additionally describes using television equipment as user equipment. *See col. 27:60-28:24.* Alternatively, *Lawler* additionally describes using television equipment as user equipment. *See fig. 1.*

In regards to claims 12 and 41: *Brenner '068* additionally describes recording a given race with a video cassette recorder. *See col. 27:65-28:15.* Alternatively, *Lawler* describes recording a given event with a video cassette recorder. *See col. 1:33-42,3:36-39.*

In regards to claim 13 and 42: *Lawler* additionally describes Recording the given race on a videocassette recorder or, alternatively, other digital recording device. *See col. 3:28-67.*

In further regards to dependent claims 14 and 43: *Brenner '068* additionally describes employing a computer as user equipment. Notably, the examiner interprets to claims use of "terminal" to include computers. *See fig. 1; col. 7:21-34.* Alternatively, *Lawler* additionally describes employing a computer as user equipment. *See fig. 2.*

In further regards to the claims 15 and 44: *Lawler* additionally describes employing telephone equipment as user equipment. *See col. 5:29-36.*

In further regards to dependent claim 16: *Brenner '068* additionally describes recording the race in real-time. *See col. 6:55-62.* Alternatively, *Lawler* additionally describes recording the events in real-time. *See fig. 7.*

In further regards to dependent claim 17 and 47: *Brenner '068* additionally describes recording the race after it has taken place. *See col. 26:65-27:22.* Alternatively, *Lawler* additionally describes recording the race after it has taken place. *See fig. 7.*

In further regards to dependent claim 18: *Brenner '068* additionally describes charging a fee for recording a given race. *See col. 27:33-39.*

In further regards to the claim 39: *Lawler* additionally describes locating control circuitry in a "set-top box". *See fig. 1.*

Response to Arguments

Applicant's arguments with respect to claims 1-19 and 37-48 have been considered but are moot in view of the new ground(s) of rejection.
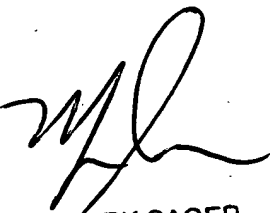
Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Steven Ashburn whose telephone number is 703 305 3543. The examiner can normally be reached on Monday thru Friday, 8:00 AM to 4:30 PM. If attempts to reach the examiner by telephone are

unsuccessful, the examiner's supervisor, Tom Hughes can be reached on 703-308-1806. The fax phone

numbers for the organization where this application or proceeding is assigned are 703 872 9302 for

regular communications and 703 872 9303 for After Final communications. Any inquiry of a general

nature or relating to the status of this application or proceeding should be directed to the receptionist

whose telephone number is 703 308 1078.

S.A.
July 15, 2003

MARK SAGER
PRIMARY EXAMINER

EVIDENCE APPENDIX C
COPY OF THE REPLY TO OFFICE ACTION DATED NOVEMBER 20, 2003

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants   :   Connie T. Marshall and Masood Garahi

Application No.:   09/609,073   Confirmation No.: 2964

Filed   :   June 30, 2000

For   :   INTERACTIVE WAGERING SYSTEMS AND
                      METHODS FOR RECORDING WAGERING EVENTS

Group Art Unit :   3714

Examiner   :   Steven Ashburn

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

PETITION UNDER 37 C.F.R. § 1.136(a)
FOR EXTENSION OF TIME AND REPLY TO OFFICE ACTION

Sir:

Responsive to the July 21, 2003 Office Action
issued in connection with the above-identified patent
application, applicants respectfully request
reconsideration of this case in light of the following
remarks.

A Listing of the Pending Claims begins on page 2 of this
Reply.

Remarks begin on page 8 of this Reply.

EV133110443US

LISTING OF THE PENDING CLAIMS

This listing of claims will replace all prior versions, and listings, of the claims in this application.

1.   (Previously presented) A method for a user at user equipment to interactively wager on races with an interactive wagering application implemented using the user equipment, comprising:

allowing the user to create and place a wager for a given race by interacting with a plurality of wager creation options;

automatically providing the user with an option to record the given race while the user is interacting with the plurality of wager creation options; and

recording the given race.

2.   (Original) The method defined in claim 1 further comprising allowing the user to select yes or no in response to the option to record the given race.

3.   (Original) The method defined in claim 2 wherein the given race is recorded in response to the user selecting yes to the option to record the given race.

4.   (Original) The method defined in claim 1 wherein the given race is recorded in a personal archive.

5.   (Original) The method defined in claim 4 wherein the personal archive is located at the user equipment.

6.   (Original) The method defined in claim 4 wherein the personal archive is located remote from the user equipment.

7.   (Original) The method defined in claim 4 further comprising using the interactive wagering application to allow the user to access the personal archive and view previously recorded races.

8.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding track name.

9.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding race number.

10.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding date.

11.   (Original) The method defined in claim 1 wherein the user equipment is user television equipment.

12.   (Original) The method defined in claim 11 wherein the given race is recorded with a videocassette recorder.

13.   (Original) The method defined in claim 11 wherein the given race is recorded with a digital video recorder.

3

14.   (Original) The method defined in claim 1 wherein the user equipment is user computer equipment.

15.   (Original) The method defined in claim 1 wherein the user equipment is user telephone equipment.

16.   (Original) The method defined in claim 1 wherein the given race is recorded in real-time.

17.   (Original) The method defined in claim 1 wherein the given race is recorded after the race has taken place.

18.   (Original) The method defined in claim 1 wherein the user is charged a fee for recording the given race.

19.   (Original) A method for a user at user equipment to interactively wager on races with an interactive wagering application implemented using the user equipment, comprising:

          allowing the user to create and place a wager for a given race;

          automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race; and

          recording the given race.

20-36.   (Cancelled)

37. (Original) An interactive wagering system in which an interactive wagering application is implemented on user equipment that provides a user with an opportunity to place wagers on races to be run, comprising:

control circuitry configured to allow the user to create and place a wager for a given race by interacting with a plurality of wager creation options, wherein while the user is interacting with the plurality of wager creation options the user is automatically provided with an option to record the given race;

a display device that displays the plurality of wager creation options; and

a recording device that records the given race.

38. (Original) The interactive wagering system defined in claim 37 wherein the user equipment is user television equipment.

39. (Original) The interactive wagering system defined in claim 38 wherein the control circuitry is located within a set-top box.

40. (Original) The interactive wagering system defined in claim 38 wherein the display device is a television.

41. (Original) The interactive wagering system defined in claim 38 wherein the recording device is a videocassette recorder.

5

42.   (Original) The interactive wagering system
defined in claim 38 wherein the recording device is a
digital video recorder.

43.   (Original) The interactive wagering system
defined in claim 37 wherein the user equipment is user
computer equipment.

44.   (Original) The interactive wagering system
defined in claim 37 wherein the user equipment is user
telephone equipment.

45.   (Original) The interactive wagering system
defined in claim 37 wherein the recording device is located
remote from the user equipment.

46.   (Original) The interactive wagering system
defined in claim 37 wherein the given race is recorded in a
personal archive.

47.   (Original) The interactive wagering system
defined in claim 46 wherein the control circuitry is
further configured to allow the user to access the
recording device and view previously recorded races.

48.   (Original) An interactive wagering system in
which an interactive wagering application is implemented on
user equipment that provides a user with an opportunity to
place wagers on races to be run, comprising:

control circuitry configured to allow the
user to create and place a wager for a given race, wherein
the user is automatically provided with an opportunity to

record the given race in response to the user placing the
wager for the given race; and

a recording device that records the given
race.

49-56. (Cancelled)

REMARKS

I. Introduction

Applicants thank the Examiner for the telephonic interview of June 3, 2003, and the Examiner's subsequent decision to reopen prosecution of this case.

Pursuant to 37 C.F.R. § 1.136(a), applicants hereby petition for a one-month extension of time to respond to the Office Action dated July 21, 2003. With the extension, a response is due on or before November 21, 2003. A check in the amount of $110.00, in payment of the fee required under 37 C.F.R. § 1.17(a)(1), is enclosed herewith. The Director is hereby authorized to charge any additional fees that may be due, or to credit overpayment of same, to Deposit Account No. 06-1075. A duplicate copy of this Reply is enclosed herewith.

Claims 1-19 and 37-48 are pending.

Claims 1-19 and 37-48 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Brenner et al., U.S. Patent No. 5,830,068 ("Brenner '068") in view of Dan Wagner et al., The Human Factors Design Guide ("HFDG"), and Lawler et al., U.S. Patent No. 5,805,763 ("Lawler").

Claims 1-19 and 37-48 were also rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over either claims 1-59 of Brenner et al., U.S. Patent No. 6,004,211 ("Brenner '211") or claims 1-132 of Brenner '068, each in view of HFDG and Lawler.

These rejections are respectfully traversed.

8

II. <u>Listing of the Pending Claims</u>

Applicants provide herewith a listing of the pending claims in this case (see pages 2-6 of this Reply). In an Amendment to claim 1 filed January 28, 2002, applicants included a correct marked-up amended claim 1, reciting the feature of "automatically providing the user with an option to record the given race ..." (See the Appendix of applicants' January 28, 2002 Reply). However, applicants have recently discovered that an incorrect claim 1, in clean format, was submitted reciting the feature of "automatically prompting the user to decide whether to record the given race ..." (see page 2 of applicants' January 28, 2002 Reply). Accordingly, in order to clarify any confusion that may have occurred due to this inadvertent error, applicants have provided herewith a correct listing of the claims.

III. <u>Applicants' Reply to the Rejections Under § 103(a)</u>

Claims 1-19 and 37-48 were rejected under 35 U.S.C. § 103 as being unpatentable over Brenner '068 in view of HFDG and Lawler. Applicants' independent claims 1, 19, 37, and 48 generally relate to systems and methods for allowing users to wager on and record wagering events. As set forth in independent claims 1 and 37, a user is allowed to "create and place a wager for a given race by interacting with a plurality of wager creation options," and is automatically provided with an option to record a race "while the user is interacting with the plurality of wager creation options." Applicants' independent claims 19 and 48 recite "allow[ing] the user to create and place a wager for a given race" and automatically providing the user with an opportunity to record the given race "in response to the user placing the wager for the given race."

As stated in the Office Action, Brenner '068 discloses an interactive wagering system that allows users to interact with a plurality of wager creation options and place bets. See, for example, FIGS. 36-39 and 41-44 of Brenner '068. Brenner '068 also shows that a selectable option associated with a given race may be presented, and that the user may select the option to record the race. See FIG. 49 and the accompanying text at column 28, lines 4-23. However, the Office Action concedes that Brenner does not show "automatically prompting the user to decide whether to record the race while interacting [with] the plurality of wager creation option." See page 14 of the Office Action.

The Office Action attempts to show this feature of applicants' claims using disclosure from Lawler.* Lawler, however, generally relates to an interactive television program guide for allowing users to browse and selecting television programs. See abstract. Lawler also shows that a selectable option (130) associated with a given television program may be presented, and that the user may select the option to record the television

---

* Applicants would like to point out that this feature is not present in applicants' pending claims. Applicants believe that the Examiner may incorrectly believe this feature to be part of applicants' claims because of applicants' inadvertent error in presenting a clean version of claim 1 in applicants' Reply to Office Action filed January 28, 2002. Instead, applicants submit that claims 1 and 37 include the feature of automatically providing the user with an option to record the given race while the user is interacting with the plurality of wager creation options, and that claims 19 and 48 include the feature of automatically providing the user with an opportunity to record a given race in response to the user placing a wager for the given race. Applicants have assumed, for the purpose of this Reply, that the Examiner would have relied on Lawler to show these features of applicants' claims.

10

program. See FIG. 6 of Lawler. When a user selects the record option in Lawler, the user is presented with additional record options. See FIG. 9. Indeed, both Lawler and Brenner '068 show similar techniques for allowing users to record a program and a race, respectively (i.e., allowing users to navigate a content selection menu that provides a selectable option for allowing users to record content). However, neither Brenner '068, Lawler, nor their combination show or suggest automatically providing the user with an opportunity to record a race "while the user is interacting with the plurality of wager creation options" or "in response to the user placing the wager for the given race" as specified by applicants' claims.

The Office Action also relies on HFDG to show these features of applicants' claims. However, HFDG is described as "a comprehensive reference tool that will help human factors professionals within the Federal Aviation Administration (FAA) and contractor organizations to efficiently carry out FAA human factors policy." See page i of the Foreword. In addition, HFDG "provides reference information to assist in the selection, analysis, design, development, and evaluation of new and modified FAA systems, facilities, and equipment." See page 1-1. Although HFDG does, as the Office Action suggest, refer to various fundamental goals for implementing human-computer interfaces, HFDG fails to refer to designing interactive wagering interfaces. Furthermore, HFDG fails to make any reference to allowing a user to interact with a plurality of wager creation options, allowing a user to place a wager, allowing a user to record wagering events, or any combination of these features of applicants' claims.

Nevertheless, the Office Action indicates that the teachings of Lawler and HFDG would have suggested to one skilled in the art to modify Brenner '068 to achieve the benefit of applicants' claimed approach. See pages 19-20 or the Office Action. Specifically, the Office Action concludes that:

> it would have been obvious ... to modify the off-track wagering system disclosed by Brenner to add the feature of automatically prompting the user to decide whether to record the race while interacting with the plurality of wager creation options.

See page 20 of the Office Action. The Examiner appears to rely on various generic and fundamental goals set forth in HFDG (e.g., for optimizing human-computer interfaces) to suggest that applicants' claims would have been obvious in light of Brenner '068 and Lawler (see page 20 of the Office Action). However, as discussed above, HFDG provides reference information for Federal Aviation Administration systems and equipment and does not refer at all to interactive wagering applications or recording wagering events. Accordingly, HDFG fails to show or suggest applicants' claimed arrangement for automatically providing a user with an opportunity to record a race "while the user is interacting with [a] plurality of wager creation options" or "in response to the user placing [a] wager for [a] given race" as specified by applicants' claims. Therefore, because neither Brenner '068, Lawler, HFDG, nor their combination show or suggest these features of applicants' claims, applicants respectfully submit that a *prima facie* case of obviousness had not been met and that the § 103 rejections should be withdrawn. See MPEP § 2143.

Moreover, the above statement of motivation provided by the Examiner is tantamount to saying that it would have been obvious to modify Brenner '068 with Lawler because it would have led to applicants' novel approach. It is well-settled, however, that to establish a *prima facie* case of obviousness, an objective reason or motivation must be provided that would lead one skilled in the art to selectively modify the prior art to arrive at applicants' claimed approach. See In re Rouffet, 149 F.3d 1350, 1357; see also In re Lee 277 F.3d 1338. Applicants respectfully submit that there is no objective evidence of record, other than applicants' disclosure, that would lead one skilled in the art to modify Brenner '068, Lawler, and HFDG to automatically provide users with an option to record a race "while the user is interacting with the plurality of wager creation options" or "in response to the user placing the wager for the given race" as specified by applicants' claims.

Without objective evidence of a motivation to modify the references to arrive at applicants' claimed approach, the Office Action "simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability," a practice that is insufficient as a matter of law. See In re Dembiczak, 175 F.3d 994, 999 (Fed. Cir. 1999); see also In re Lee at 1344 ("[i]t is improper, in determining whether a person of ordinary skill would have been led to a combination of references, simply to use that which the inventor taught against its teacher"). For at least this reason, applicants respectfully request that the § 103 rejections should be withdrawn.

IV. Applicants' Reply to the Double Patenting Rejections

Claims 1-19 and 37-48 were rejected under the judicially created doctrine of obviousness-type double patenting (analogous to a rejection under 35 U.S.C. § 103 according to MPEP § 804(II)(B)(1)) as being unpatentable over either claims 1-59 of Brenner '211 or claims 1-132 of Brenner '068, each in view of HFDG and Lawler. Applicants respectfully submit, however, that the obviousness-type double patenting rejections are improper in this case.

It is well settled that, in cases where double patenting may be at issue, "it must always be carefully observed that the ... patent [used as the basis for a double patenting rejection] is not 'prior art' under either section 102 or section 103 of the 1952 Patent Act (35 U.S.C. as amended)." In re Boylan, 392 F.2d 1017, 1018; see also In re Braithwaite, 379 F.2d 594, 600, n.4 ("While analogous to the non-obviousness requirement of 35 U.S.C. § 103, that section is not itself involved in double patenting rejections because the patent principally underlying the rejection is not prior art"). Indeed, the courts have determined that double patenting rejections are reserved for situations "where patents are not citable as a reference against each other and therefore can not be examined for compliance with the rule that only one patent is available per invention." Eli Lilly & Co. v. Barr Labs., 251 F.3d 955, 966 (Circuit Judge Newman dissenting, in a separate opinion, on the Court's refusal to reconsider the case en banc); see also General Foods Corp. v. Studiengesellschaft Kohle mbH, 972 F.2d 1272, 1278.

In the present case, Brenner '068 was filed on September 8, 1995 and issued on November 3, 1998, which is

before the date of applicants' claimed invention.* Accordingly, the claims and disclosure of Brenner '068 are statutory prior art under 35 U.S.C. § 102(a). Therefore, the double patenting rejections based on the claims of Brenner '068 are improper. See MPEP § 804. Moreover, Brenner '211 claims priority from Brenner '068 and the claims of Brenner '211 are therefore fully supported by the disclosure of Brenner '068. Accordingly, the obviousness-type double patenting rejections based on the subject matter of claims 1-59 of Brenner '211 are also improper because this subject matter is fully supported by Brenner '068, which is statutory prior art under 35 U.S.C. § 102(a). For at least these reasons, the obviousness-type double patenting rejections should be withdrawn.

However, even if the double-patenting rejections are found to be proper, applicants respectfully submit that, for the same reasons set forth above in connection with the rejections under § 103, applicants' claims 1-19 and 37-48 are not obvious in view of claims 1-132 of Brenner '068 and claims 1-59 of Brenner '211, each in view of HFDG and Lawler.

---

* Applicants' non-provisional patent application was filed on January 30, 2000 and claims priority from U.S. provisional patent application No. 60/142,174, filed July 1, 1999.

V.   Conclusion

The foregoing demonstrates that the obviousness-type rejections of claims 1-19 and 37-48 should be withdrawn. This application is therefore in condition for allowance. Reconsideration and allowance of this application are respectfully requested.

Respectfully submitted,

James A. Leiz
Registration No. 46,109
Agent for Applicants
FISH & NEAVE
Customer No. 1473
1251 Avenue of the Americas
New York, New York 10020-1104
Tel.: (212) 596-9000

EVIDENCE APPENDIX D
COPY OF THE FINAL OFFICE ACTION DATED FEBRUARY 11, 2004

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/609,073	06/30/2000	CONNIE T MARSHALL	ODS-9	2964

1473   7590   02/11/2004

FISH & NEAVE
1251 AVENUE OF THE AMERICAS
50TH FLOOR
NEW YORK, NY 10020-1105

EXAMINER
ASHBURN, STEVEN L

ART UNIT	PAPER NUMBER
3714	

DATE MAILED: 02/11/2004   22

Please find below and/or attached an Office communication concerning this application or proceeding.

RECEIVED

FEB 1 8 2004

FISH & NEAVE - PATENT DEPT.
REFERRED TO _____
NOTED BY _____

REMINDER - PLEASE INITIAL
THIS DOCUMENT TO INDICATE
THAT YOU HAVE SEEN IT

DOCKETED FOR 5/11/04

FINAL

	Application No.	Applicant(s)
Office Action Summary	09/609,073	MARSHALL ET AL.
	Examiner	Art Unit
	Steven Ashburn	3714

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _20 November 2003_.

2a)☒ This action is **FINAL.**   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-19 and 37-48_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-19 and 37-48_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Double Patenting

Claims 1-19 and 37-48 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over either claims 1-59 of Brenner et al., U.S. Patent 6,004, 211 (Dec. 21, 1999) (hereinafter "*Brenner '211*").

This holding, incorporated herein, is maintained from the prior action for the cited claims. Response to the applicant's remarks are provided below and incorporated herein.

Claim Rejections - 35 USC § 103

Claims 1-19 and 37-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brenner, U.S. 5,830,068 (Nov. 3, 1998) in view of Dan Wagner et al., 'The Human Factors Design Guide', DOT/FAA/CT-96/1 (Jan. 15, 1996) (hereinafter "*HFDG*") and Lawler et al., U.S. 5,805,763 (Sep. 8, 1998).

This holding, incorporated herein, is maintained from the prior action for the cited claims. Response to the applicant's remarks are provided below and incorporated herein.

Response to Arguments

Applicant's arguments filed Nov. 20, 2003 with respect to the obviousness-type double patenting rejection of claims 1-19 and 37-48 in view of *Brenner '068* are persuasive. *See pp. 14-15*. Therefore, the rejection has been withdrawn.

Applicant's arguments filed Nov. 20, 2003 with respect to the obviousness-type double patenting rejection of claims 1-19 and 37-48 in view of *Brenner '211* are not persuasive. The applicant asserts that

the rejection is improper because the features claimed in *Brenner '211* are fully supported by the

disclosure of *Brenner '068* patent. The examiner respectfully disagrees.

A rejection based on nonstatutory double patenting is based on a judicially created doctrine

grounded in public policy so as to prevent the unjustified or improper timewise extension of the right to

exclude granted by a patent. *In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re*

Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686 F.2d 937, 214 USPQ 761

(CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); In re Thorington, 418 F.2d

528, 163 USPQ 644 (CCPA 1969); In re White, 405 F.2d 904, 160 USPQ 417 (CCPA 1969); In re

Schneller, 397 F.2d 350, 158 USPQ 210 (CCPA 1968); In re Sarett, 327 F.2d 1005, 140 USPQ 474

(CCPA 1964). The A double patenting rejection of the obviousness-type is "analogous to [a failure to

meet] the nonobviousness requirement of 35 U.S.C. 103" except that the patent principally underlying

the double patenting rejection is not considered prior art. *In re Braithwaite, 379 F.2d 594, 154 USPQ 29*

(CCPA 1967). Therefore, any analysis employed in an obviousness-type double patenting rejection

parallels the guidelines for analysis of a 35 U.S.C. 103 obviousness determination. *In re Braat, 937 F.2d*

589, 19 USPQ2d 1289 (Fed. Cir. 1991); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985).

MPEP 804-II-B-1 states the proper method analysis for non-statutory, obviousness-type double patenting:

> ... the analysis employed in an obviousness-type double patenting determination parallels
> the guidelines for a 35 U.S.C. 103(a) rejection, the factual inquiries set forth in Graham
> v. John Deere Co., 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a
> background for determining obviousness under 35 U.S.C. 103 are employed when
> making an obvious-type double patenting analysis. These factual inquiries are
> summarized as follows:
>> (A)   Determine the scope and content of a patent <u>claim</u> and the prior art relative to a
>> <u>claim</u> in the application at issue;
>> (B)   Determine the differences between the scope and content of the patent claim
>> and the prior art as determined in (A) and the claim in the application at issue;
>> (C)   Determine the level of ordinary skill in the pertinent art; and
>> (D)   Evaluate any objective indicia of nonobviousness.
> The conclusion of obviousness-type double patenting is made in light of these factual
> determinations (emphasis added).

As described in the MPEP, obviousness-type double patenting is determined based on a comparison of claims, not disclosures. A single disclosure may support several independent inventions. Hence, the fact that two inventions share a common disclosure is of no matter in determining obvious-type double patenting.

In this case, the applicant did not file a terminal declaimer limiting the term *Brenner '211* to the duration of *Brenner '068*. Hence, despite their common disclosures, the inventions claimed in *Brenner '211* and *Brenner '068* are independent. Consequently, a nonstatutory double patenting rejection in view of *Brenner '211* is proper is to prevent the unjustified or improper timewise extension of the right to exclude granted by the patent.

Applicant's arguments filed Nov. 20, 2003 with respect to the rejection under *35 USC § 103* have been fully considered but they are not persuasive. The examiner's response is provide below.

First, the applicant's note that the feature "automatically prompting the user to decide whether to record the race while interacting with the plurality of wager creation options", as stated in the examiner's rejection, is not present in pending claims. Instead, the applicants submit that claims 1 and 37 include the feature of automatically providing the user with an option to record a given race while the user in interacting with a plurality of wager creation options. Furthermore, the applicants submit that claim 19 and 48 include the feature of automatically providing the user with an opportunity to record a given race in response to the user placing a wager for the given race. The examiner concurs. Regardless of the differences in language, the rejection is maintained from the prior action for the reasons discussed below

Second, the applicant argues that the pending claims distinguish over the prior art because the gaming device described by the combination of *Brenner '068* with *HFDG* and *Lawler* does not suggest the features of automatically providing the user with an opportunity to record a given race (i) while the

user in interacting with a plurality of wager creation options or (ii) in response to the user placing a wager

for the given race. The examiner respectfully disagrees. The standard of patentability is what the prior

art, taken as a whole, suggests to an artisan at the time of the invention. *In re Merck & Co., Inc.,* *800*

F.2d 1091, 097, 231 USPQ 375, 379 (Fed. Cir. 1986). The question is not only what the references

expressly teach, but what they would collectively suggest to one of ordinary skill in the art. *In re Simon,*

461 F.2d 387, 1390, 174 USPQ 114, 116 (CCPA 1972).

In this case, *Brenner '068* discloses an "off-track" wagering system that allows users to create

and place wagers by interacting with an interactive wager-creation interface and provides users with an

option to record a race while the user is interacting with a plurality of wager-creation options. *See fig. 31-*

36. The option to record a race is be selected from the sub-menu shown in fig. 34(596) rather than being

displayed automatically as part of the wager-creation menu shown in fig. 31(448).

Lawler discloses a system that is analogous to *Brenner '068* in many ways. Similar to *Brenner*

'068, wherein users browse race information and place orders to record racing videos, *Lawler* allows

users to browse entertainment program information and place orders to record event videos. *See fig. 3;*

col. 1:45-2:40. Furthermore, both systems are embodied on "set-top" boxes used to control television

programming transmitted across broadcast networks. *See Lawler, fig. 1(18); col. 3:45-67. See Brenner*

'068, fig.1, col. 3:48-4:9. Finally, both systems allow users the opportunity to record an event or set a

reminder. *See Lawler, fig. 8, 9. See Brenner '068, col. 4:57-67.*

In regard to the claimed features, *Lawler* automatically provides users with an opportunity to

record a given event (i) while the user in interacting with a plurality of menu options or (ii) in response to

the user placing an order for a given item. *See fig. 4(a)(b), 6-10.* More specifically, it illustrates that a

user is automatically given the option to record a program in response to an order while the user is

interacting with the menu options. *See fig. 4A(222, 226, 224, 236, 238), 6(102, 136).* As a result, users

are able to quickly and easily identify and select events using an interactive user-interface and to

designate the selected event for recording. *See col. 1:45-50; col. 13:38-41.*

The *HFDG* provides guidance in the design of human-computer interfaces. It defines knowledge

that is within the ordinary skill of artisans who work in the field of human-computer interfaces. Gaming

artisan fall within this class because a primary element of wagering devices is the interface between a

device and the player. Hence, the methods described in the *HFDG* would be within the ordinary

knowledge of a gaming artisan.

As listed in the previous action, the *HFDG* suggests many features relevant to the pending claims.

In particular, the teachings include the following:

a.   Prompting control entries: a system or application shall provide the user whatever

information is required to guide control entries. *See 8.1.6.3.* For example, prompts may be

incorporated into a display at <u>any point</u> in a transaction sequence that will be helpful, or prompts

may appear in response to a request for help. *See id.* Notably, this suggests automatically

prompting users with control entries. In addition, it suggests that is it a matter of design choice

when to have the prompts occur.

b.   Display of all options: a menu shall display explicitly and completely all options

available to a user at the current step in a transaction sequence. *See 8.1.11.1.7.*

c.   Easy selection of important options: hierarchical menus should permit immediate user

access to critical or frequently selected options. *See 8.1.11.3.5.*

The methods listed above suggest to an artisan to modify the user interface disclosed in *Brenner '068* to

provide a more effective user-interface. For example, in the system disclosed by *Brenner '068* wherein

the wager-creation step is associated with additional options available in sub-menu, it suggest modifying

the wager-creation menu to display all the available options. In addition, the *HFDG* suggests modifying

the wager-creation interface to give immediate access frequently used functions.

Thus, when the prior art is taken as a whole at a time prior to the invention it collectively suggest

an "off-track" wagering system that automatically provides the user with an opportunity to record a given

race while the user in interacting with a plurality of wager creation options or in response to the user

placing a wager for the given race. As suggested by *Lawler*, the modification would enhance the user-

interface disclosed by *Brenner '068* by providing a reliable an easy method to record events for users who

are otherwise unable to program the system. Furthermore, as suggested by the *HFDG*, automatically

providing options in systems such as disclosed by *Brenner '068* improves the user-interface provides

users whatever information is required to guide control entries by explicitly and completely displaying all

options available to a user at the current step in a transaction sequence, permitting immediate user access

to critical or frequently selected options, and eliminating the need for the user to remember the options.

Third, the applicant appears to assert that *Lawler* does not describe automatically providing the

user with an option to record content. The examiner respectfully disagrees. The reference clearly

illustrates that a user is automatically given the option to record a program in response to an order while

the user is interacting with the menu options. *See fig. 4A(222, 226, 224, 236, 238), 6(102, 136)*. As

discussed in the immediately above, the *HFDG* suggests automatically prompting users with options

associated with a display.

Fourth, the applicant argues that *Lawler* and *HFDG* are not directed to interactive wagering

devices or interacting with a wager-creation options. In response, it has been held that a prior art

reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the

particular problem with which the applicant was concerned, in order to be relied upon as a basis for

rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992).

On its face, the argument that the prior art is non-analogous because it is not directed toward

wagering devices or wagering transactions in wholly unpersuasive. An idea is not cordoned-off from the

whole of human knowledge simply because it is directed toward wagering. Wagering devices are not an

isolated art. Instead, they are complex systems comprised of many technologies from arts including, but

not limited to, amusement devices, data processing, networking, computer graphics processing, electronic

fund transfer, interactive video distribution, check-actuated control mechanisms, article dispensing,

merchandising and cryptography. In this case, the claimed invention is a type of interactive video

distribution system modified to offer wagering services. The wagering aspect does not delineate the

system from interactive video distribution systems. Wagering services are merely a type of consumer

business transaction. Whereas a typical transaction provides goods/services in exchange for

consideration, a wagering transaction provides the service of a game of chance in exchange for

consideration of a wager. Hence, the pending claims do not distinguish over the prior art of record

because the service offered is a game of chance or because they involve a wagering transaction.

In regard to *Lawler*, the reference discloses system is that analogous to *Brenner '068* in many

ways. In *Brenner '068* users collect race information and place orders to record racing videos. Similarly,

Lawler allows users to collect entertainment program information and place orders to record event videos.

See fig. 3; col. 1:45-2:40. Furthermore, both systems are embodied on "set-top" boxes used to control

television programming transmitted across broadcast networks. *See Lawler, fig. 1(18); col. 3:45-67. See*

Brenner '068, fig.1, 2; col. 3:48-4:9. In addition, both systems allow users the opportunity to record an

event or set a reminder. *See Lawler, fig. 8, 9. See Brenner '068, col. 4:57-67.* Hence, *Lawler* is

analogous because the reference is in the field of the applicant's endeavor of providing an interactive

user-interface. Also, *Lawler* is analogous because it is reasonably pertinent to the particular problem of

simplifying the process of selecting a option to record an event though a interactive user-interface.

In regard to the *HFDG*, the reference teaches fundamental considerations for designing human-computer interfaces. Thus, its teachings are applicable to wagering devices because wagering devices require the similar considerations as human-computer interfaces found in other devices. Hence, the *HFDG* is reasonably pertinent to the particular problem of improving interactive user-interfaces.

Hence, for the reasons given above the examiner maintains that the prior art is analogous to the claimed invention.

Fifth, the applicant argues that there is no suggestion to combine the references. The examiner respectfully disagrees. "There are three possible sources for a motivation to combine references: the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art." *In re Rouffet, 149 F.3d 1350, 1357, 47 USPQ2d 1453, 1457-58 (Fed. Cir. 1998).*

In this case, *Lawler* discloses an analogous interactive user interface for an entertainment system allowing users to search and automatically record broadcast events. *See fig. 3; col. 1:45-2:40.* In particular, *Lawler* discloses a recording system allows users to quickly and easily select an event to record from an interactive menu and thereby solves the problem of users who remain unable to program their systems. *See col. 1:40-43, col. 13:38-42.* Hence, *Lawler* suggests the modification of *Brenner '068* because it is directed to the problem to be solved and explicitly suggests the modification.

Furthermore, the *HFDG* defines knowledge within the ordinary skill of an artisan who works in the field of human-computer interfaces. Moreover, it provides motivations for incorporating various design features into the interfaces. As discussed previously , the *HFDG* suggests improving user-interfaces by providing users whatever information is required to guide control entries, explicitly and completely displaying all options available to a user at the current step in a transaction sequence, permitting immediate user access frequently selected options, and eliminating the need for the user to remember the entries. *See supra.* Hence, the *HFDG* suggests the modification of *Brenner '068* based on

the nature of the problems to be solved in interactive user-interfaces, the teachings of the *HFDG* with

respect to providing improved interactive user-interfaces, and the knowledge of persons of ordinary skill

in the art as defined by the *HFDG*.

Hence, when the disclosures of *Lawler* and the *HFDG* are taken as a whole by one of ordinary

skill in the art of gaming devices, they collectively suggest modifying *Brenner '068* to add the feature of

automatically provides the user with an opportunity to record a given race while the user in interacting

with a plurality of wager creation options or in response to the user placing a wager for the given race.

Moreover, this suggestion is found completely within the prior art and not based upon the applicant's

disclosure.

Consequently, for all the reasons given above, the rejection of the pending claims is respectfully

maintained.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set

forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire

THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end

of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the

date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated

from the mailing date of the advisory action. In no event, however, will the statutory period for reply

expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should

be directed to Steven Ashburn whose telephone number is 703 305 3543. The examiner can normally be

reached on Monday thru Friday, 8:00 AM to 4:30 PM. If attempts to reach the examiner by telephone are

unsuccessful, the examiner's supervisor, Tom Hughes can be reached on 703-308-1806. The fax phone

number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application

Information Retrieval (PAIR) system. Status information for published applications may be obtained

from either Private PAIR or Public PAIR. Status information for unpublished applications is available

through Private PAIR only. For more information about the PAIR system, see http://pair-

direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

s.a.

MARK SAGER
PRIMARY EXAMINER

EVIDENCE APPENDIX E
COPY OF THE REPLY TO FINAL OFFICE ACTION
DATED JULY 12, 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants   :   Connie T. Marshall et al.

Application No.:   09/609,073   Confirmation No.: 2964

Filed   :   June 30, 2000

For   :   INTERACTIVE WAGERING SYSTEMS AND METHODS FOR
                        RECORDING WAGERING EVENTS

Group Art Unit :   3714

Examiner   :   Steven L. Ashburn

Mail Stop AF
Hon. Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

REPLY TO FINAL OFFICE ACTION

Sir:

This reply is in response to the February 11, 2004 Final Office Action issued in connection with the above-identified patent application. Please grant a two-month extension of time under 37 C.F.R. § 1.136(a) to the February 11, 2004 Office Action. With the extension, the time for replying is extended up to and including July 12, 2004 (owing to July 11, 2004 being a Sunday). A check in the amount of $420.00, in payment of the fee set forth in 37 C.F.R. §1.17(a)(2) is enclosed.

Applicants respectfully request reconsideration of this case in light of the following remarks.

AMENDMENTS TO THE CLAIMS begins on page 2 of this paper.

REMARKS begin on page 7 of this Reply.

Express Mail:
EV371752324US

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application. In this listing of claims, claims 2, 4, 11, 14-18, 38 and 43-46 have been amended. Claims 1 and 37 have been cancelled.

Listing of Claims:

1.   (Cancelled)

2.   (Currently Amended) The method defined in claim [[1]] 19 further comprising allowing the user to select yes or no in response to the option to record the given race.

3.   (Original) The method defined in claim 2 wherein the given race is recorded in response to the user selecting yes to the option to record the given race.

4.   (Currently Amended) The method defined in claim [[1]] 19 wherein the given race is recorded in a personal archive.

5.   (Original) The method defined in claim 4 wherein the personal archive is located at the user equipment.

6.   (Original) The method defined in claim 4 wherein the personal archive is located remote from the user equipment.

7.   (Original) The method defined in claim 4 further comprising using the interactive wagering application to allow the user to access the personal archive and view previously recorded races.

8.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding track name.

9.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding race number.

10.   (Original) The method defined in claim 7 wherein the previously recorded races in the personal archive are listed with their corresponding date.

11.   (Currently Amended) The method defined in claim [[1]] 19 wherein the user equipment is user television equipment.

12.   (Original) The method defined in claim 11 wherein the given race is recorded with a videocassette recorder.

13.   (Original) The method defined in claim 11 wherein the given race is recorded with a digital video recorder.

14. (Currently Amended) The method defined in claim [[1]] 19 wherein the user equipment is user computer equipment.

15. (Currently Amended) The method defined in claim [[1]] 19 wherein the user equipment is user telephone equipment.

16. (Currently Amended) The method defined in claim [[1]] 19 wherein the given race is recorded in real-time.

17. (Currently Amended) The method defined in claim [[1]] 19 wherein the given race is recorded after the race has taken place.

18. (Currently Amended) The method defined in claim [[1]] 19 wherein the user is charged a fee for recording the given race.

19. (Original) A method for a user at user equipment to interactively wager on races with an interactive wagering application implemented using the user equipment, comprising:

allowing the user to create and place a wager for a given race;

automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race; and

recording the given race.

20-37. (Cancelled)

4

38. (Currently Amended) The interactive wagering system defined in claim [[37]] 48 wherein the user equipment is user television equipment.

39. (Original) The interactive wagering system defined in claim 38 wherein the control circuitry is located within a set-top box.

40. (Original) The interactive wagering system defined in claim 38 wherein the display device is a television.

41. (Original) The interactive wagering system defined in claim 38 wherein the recording device is a videocassette recorder.

42. (Original) The interactive wagering system defined in claim 38 wherein the recording device is a digital video recorder.

43. (Currently Amended) The interactive wagering system defined in claim [[37]] 48 wherein the user equipment is user computer equipment.

44. (Currently Amended) The interactive wagering system defined in claim [[37]] 48 wherein the user equipment is user telephone equipment.

45. (Currently Amended) The interactive wagering system defined in claim [[37]] 48 wherein the recording device is located remote from the user equipment.

46. (Currently Amended) The interactive wagering system defined in claim [[37]] 48 wherein the given race is recorded in a personal archive.

47. (Original) The interactive wagering system defined in claim 46 wherein the control circuitry is further configured to allow the user to access the recording device and view previously recorded races.

48. (Original) An interactive wagering system in which an interactive wagering application is implemented on user equipment that provides a user with an opportunity to place wagers on races to be run, comprising:

control circuitry configured to allow the user to create and place a wager for a given race, wherein the user is automatically provided with an opportunity to record the given race in response to the user placing the wager for the given race; and

a recording device that records the given race.

49-56. (Cancelled)

REMARKS

I. Summary of Office Action

Claims 1-19 and 37-48 were pending in this application.

Claims 1-19 and 37-48 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Brenner et al., U.S. Patent No. 5,830,068 ("Brenner '068") in view of Dan Wagner et al., The Human Factors Design Guide ("HFDG"), and Lawler et al., U.S. Patent No. 5,805,763 ("Lawler").

Claims 1-19 and 37-48 have been rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-59 of Brenner et al., U.S. Patent No. 6,004,211 (hereinafter "Brenner '211").

II. Summary of Telephonic Interview

The Examiner and the undersigned conducted a telephonic interview on June 16, 2004. The undersigned wishes to thank the Examiner for the courtesies extended during the interview.

During the interview, the rejection of the claims in the above-identified patent application were discussed. More specifically, as noted in the Interview Summary, independent claims 1 and 37 were discussed and an agreement was not reached.

III.   Summary of Applicants' Reply to Office Action

Claims 1 and 37 have been cancelled without prejudice because an agreement could not be reached during the interview. Accordingly, applicants have cancelled claims 1 and 37 and will continue prosecuting independent claims 19 and 48 in order to advance prosecution. Claims 2-18 and 38-47 were previously dependent on claims 1 and 37, respectively. Claims 2, 4, 11, 14-18, 38 and 43-46 have been amended such that claims 2-18 and 38-47 are now dependent on claims 19 and 48, respectively.

Applicants respectfully submit that the subject matter of amended claims 2, 4, 11, 14-18, 38 and 43-46 are fully supported by the originally-filed specification. No new subject matter has been added.

The Examiner's rejections are respectfully traversed.

IV.   Applicants' Reply to the Rejections Under § 103(a)

Claims 2-19 and 38-48 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Brenner '068 in view of HFDG and Lawler. Applicants' independent claims 19 and 48 generally relate to systems and methods for allowing users to wager on and record wagering events. A user is allowed to create and place a wager for a given race and is automatically provided with an opportunity to record the given race in response to the user placing the wager for the given race. The given race is then recorded.

8

A. **Brenner, Lawler And HFDG Fail To Show Or
Suggest Automatically Providing A User
With An Opportunity To Record A Race In
Response To Placing A Wager on That Race**

As stated in the Office Action, Brenner '068 discloses

an interactive wagering system that allows users to create and

place wagers by interacting with an interactive wager-creation

interface (see Brenner '068, FIGS. 36-39 and 41-44).

Brenner '068 also shows that a selectable option associated with

a given race may be presented, and that the user may select the

option to record the race (see Brenner '068, column 28,

lines 4-23 and FIG. 49). However, the Examiner concedes that

Brenner does not show automatically providing the user with an

opportunity to record the given race in response to the user

placing the wager for the given race (see Office Action,

page 5). The Office Action attempts to show this feature of

applicants' claims using the disclosure from Lawler.

Lawler generally relates to an interactive television

program guide for allowing users to browse and select television

programs (see Lawler, abstract). The Examiner contends that

Lawler automatically provides users with an opportunity to

record a given event in response to the user placing an order

for a given item in FIGS. 4A, 4B and 6-10 (see Office Action,

page 5). Applicants respectfully disagree. Program options

menu 136 of FIG. 6 includes both order button 138 and record

button 130. Applicants submit that these two buttons provide

9

distinct options for a user, and that in response to the selection of order button 138, the option to record a program is not automatically provided to the user, as contended by the Examiner. The flow diagram in FIG. 7 of Lawler illustrates this point.

FIG. 7 shows the steps taken if one of the buttons in program options menu 136 of FIG. 6 is selected. More specifically, if order button 138 in program options menu 136 is selected, the system proceeds to block 306 where a menu to facilitate ordering is then displayed at block 308. At block 310, the system monitors and implements the user's selections from the ordering menu and then at block 312 the system returns to the program time guide of FIG. 3. Nowhere in this process is it shown or suggested that the user is automatically provided with an opportunity to record a program as the Examiner contends. Therefore, Lawler does not show or suggest automatically providing users with an opportunity to record a program in response to the user placing an order for the program, as the Examiner contends. Thus, since Lawler does show or suggest automatically providing users with an opportunity to record a program in response to the user placing an order for the program, it cannot show or suggest "automatically providing the user with an opportunity to record the given race in

response to the user placing the wager for the given race," as required in applicants' independent claims 19 and 48.

The Office Action appears to rely on HFDG to provide the motivation to combine Brenner with Lawler, but it also relies on three sections in HFDG to modify the user interface disclosed in Brenner '068 to show features of applicants' claims. The Examiner first refers to section 8.1.6.3 of HFDG which states that "[t]he system or application shall provide the user whatever information is required to guide control entries." As an example, section 8.1.6.3 states that "[p]rompts may be incorporated into a display at any point in a transaction sequence that will be helpful." (HFDG, page 8-14, emphasis added). Applicants respectfully submit that this section merely shows, in a broad sense, that it would be helpful to provide prompts. However, nowhere is it shown or suggested in this section that it would be helpful to automatically provide the user with an opportunity to record a race in response to a user placing a wager for the race.

The Examiner also points to sections 8.1.11.1.7 and 8.1.11.3.5 of HFDG, which refer to providing menu options, in an attempt to show features of applicants' claims. More specifically, section 8.1.11.1.7 states that menus should display all options that are available to a user at a step in a transaction sequence (see HFDG, page 8-20). Section 8.1.11.3.5

11

states that critical or frequently selected options should be easily accessible to a user (see HFDG, page 8-22). The Examiner contends that HFDG suggests modifying Brenner '068 to display all available options and to give access to frequently used functions (see Office Action, page 6). However, applicants submit that the Office Action has not shown or suggested that providing the user with an opportunity to record a race in response to placing a wager is an available option in the transaction sequence. Furthermore, applicants submit that the Office Action has not shown or suggested that applicants' claimed feature is critical or a frequently used function. In fact, applicants' claim requires providing the user with the opportunity to record a race in response to the user placing a wager for the race. If recording a race is considered critical or frequently used, then it would seem that this feature should be part of the wager creation process, and not displayed in response to placing a wager.

Therefore, at least because Brenner, Lawler and HFDG do not show or suggest "automatically providing the user with an opportunity to record the given race in response to the user placing the wager for the given race," applicants respectfully submit that the rejection of independent claims 19 and 48 under 35 U.S.C. § 103(a) should be withdrawn.

B. The Office Action Failed To Provide Sufficient Motivation To Combine

The Office Action has failed to provide sufficient motivation for combining the references to justify the assertion of a § 103 rejection. In re Rouffet, 149 F.3d 1350, 1355 (Fed. Cir. 1998) ("When a rejection depends on a combination of prior art references, there must be some teaching, suggestion, or motivation to combine the references"); see also MPEP § 2142 and 2143.01. It is well-settled that an Office Action can "satisfy this burden only by showing some objective teaching ... that would lead [one of ordinary skill in the art] to combine the relevant teachings of the references." In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988).

As mentioned above, the Office Action conceded that Brenner does not automatically provide the user with an opportunity to record the given race in response to the user placing the wager for the given race. It attempted to the modify Brenner '068 to include this feature by relying on broad teaching in Lawler of automatically providing users with an opportunity to record a given event in response to the user placing an order for an item (see Office Action, page 6). However, as demonstrated above, Lawler does not show or suggest this teaching. Therefore, Lawler lacks the requisite motivation to modify Brenner '068 to show or suggest with objective evidence applicants' claimed feature.

13

As further demonstrated above, the sections of HFDG cited in the Office Action failed to show or suggest automatically providing the user with an opportunity to record a race in response to a user placing a wager for the race. In fact, some of the other sections of HFDG, when taken as a whole, teach away from applicants' claimed feature. For example, section 8.1.14.2.2 of HFDG states that only information that is relevant to a task should be included in a system or application (see HFDG, page 8-32). However, nowhere is it shown or suggested that providing the user with the opportunity to record a race is relevant to the task of placing a wager on the race. Therefore, HFDG would seem to suggest to not provide the user with the opportunity to record a race in response to the user placing a wager on the race.

Another example is found in section 8.1.14.4.2 of HFDG which states that a user's effort should be minimized by reducing the number of keystrokes required of users (see HFDG, page 8-33). Applicants' claims require providing the user with the opportunity to record the race in response to placing a wager. This may add keystrokes, however, which is discouraged by HFDG. If HFDG suggests minimizing keystrokes, then HFDG would seem to suggest that the user should be provided with the opportunity to record the race while placing a wager, not in response to placing the wager.

These two exemplary teachings, illustrate how HFDG teaches away from applicants' claimed approaches. In addition, applicants respectfully submit that there is no objective evidence of record, other than applicants' disclosure, that would lead one skilled in the art to modify Brenner '068, Lawler and HFDG to automatically provide users with an option to record a race "in response to the user placing the wager for the given race" as specified by applicants' claims. Without objective evidence of a motivation to modify the references to arrive at applicants' claimed approach, the Office Action "simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability," a practice that is insufficient as a matter of law. See In re Dembiczak, 175 F.3d 994, 999 (Fed. Cir. 1999); see also In re Lee at 1344 ("[i]t is improper, in determining whether a person of ordinary skill would have been led to a combination of references, simply to use that which the inventor taught against its teacher").

Therefore, because neither Brenner '068, Lawler, HFDG, nor their combination show or suggest applicants' claimed approach, applicants respectfully submit that a *prima facie* case of obviousness had not been met and that the rejections of independent claims 19 and 48 under 35 U.S.C. § 103(a) should be withdrawn. See MPEP § 2143.

Claims 2-18 are dependent from claim 19 and are allowable at least because claim 19 is allowable. Claims 38-47 are dependent from claim 48 and are allowable at least because claim 48 is allowable.

V.   Applicants' Reply to the Double Patenting Rejections

Claims 2-19 and 38-48 were rejected under the judicially created doctrine of obviousness-type double patenting (analogous to a rejection under 35 U.S.C. § 103 according to MPEP § 804(II)(B)(1)) as being unpatentable over claims 1-59 of Brenner '211. Applicants respectfully submit, however, that the obviousness-type double patenting rejection is improper.

It is well settled that, in cases where double patenting may be at issue, "it must always be carefully observed that the ... patent [used as the basis for a double patenting rejection] is not 'prior art' under either section 102 or section 103 of the 1952 Patent Act (35 U.S.C. as amended)." In re Boylan, 392 F.2d 1017, 1018; see also In re Braithwaite, 379 F.2d 594, 600, n.4 ("While analogous to the non-obviousness requirement of 35 U.S.C. § 103, that section is not itself involved in double patenting rejections because the patent principally underlying the rejection is not prior art"). Indeed, the courts have determined that a double patenting rejection is reserved for situations "where patents are not citable as a reference against each other and therefore can not

be examined for compliance with the rule that only one patent is available per invention." <u>Eli Lilly & Co. v. Barr Labs.</u>, 251 F.3d 955, 966 (Circuit Judge Newman dissenting, in a separate opinion, on the Court's refusal to reconsider the case en banc); see also <u>General Foods Corp. v. Studiengesellschaft Kohle mbH</u>, 972 F.2d 1272, 1278.

In the present case, Brenner '068 was filed on September 8, 1995 and issued on November 3, 1998, which is before the date of applicants' claimed invention.* Accordingly, the claims and disclosure of Brenner '068 are statutory prior art under 35 U.S.C. § 102(a). The Examiner agreed with the applicants' arguments in the November 20, 2003 Reply to Office Action that the double patenting rejection based on the claims of Brenner '068 was improper and withdrew the rejection (see Office Action, page 2).

The Examiner, however, maintained the double patenting rejection with respect to Brenner '211, which was filed on August 24, 1998, issued on December 21, 1999 and claims priority from Brenner '068. The Examiner contended that "obviousness-type double patenting is determined based on comparison of claims, <u>not</u> disclosures. . . the fact that two inventions share

---

* Applicants' non-provisional patent application was filed on January 30, 2000 and claims priority from U.S. provisional patent application No. 60/142,174, filed July 1, 1999.

a common disclosure is no matter in determining obviousness-type double patenting" (Office Action, page 4). However, Brenner '211 claims priority from Brenner '068 and the subject matter of the claims in Brenner '211 is fully supported and disclosed in Brenner '068. Therefore, applicants submit that the obviousness-type double patenting rejection based on the subject matter of claims 1-59 of Brenner '211 is improper because this subject matter is disclosed in Brenner '068 which is available as prior art under 35 U.S.C. § 103. Accordingly, the obviousness-type double patenting rejections should be withdrawn.

However, even if the double-patenting rejection is found to be proper, applicants respectfully submit that, for the same reasons set forth above in connection with the rejection under § 103 over Brenner '068, applicants' claims 2-19 and 38-48 are not obvious in view of claims 1-59 of Brenner '211.

VI. Conclusion

The foregoing demonstrates that the obviousness-type rejections of claims 2-19 and 38-48 should be withdrawn. This application is therefore in condition for allowance. Reconsideration and allowance of this application are respectfully requested.

Respectfully submitted,

James A. Leiz
Registration No. 46,109
Agent for Applicants
FISH & NEAVE
Customer No. 1473
1251 Avenue of the Americas
New York, New York 10020-1105
Tel.: (212) 596-9000

EVIDENCE APPENDIX F
COPY OF THE ADVISORY ACTION DATED AUGUST 12, 2004

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/609,073	06/30/2000	CONNIE T MARSHALL	ODS-9	2964

1473	7590	08/12/2004	EXAMINER

FISH & NEAVE
1251 AVENUE OF THE AMERICAS
50TH FLOOR
NEW YORK, NY 10020-1105

EXAMINER
ASHBURN, STEVEN L

ART UNIT	PAPER NUMBER
3714	

DATE MAILED: 08/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

RECEIVED

AUG 1 6 2004

FISH & NEAVE - PATENT DEPT.
REFERRED TO _____
NOTED BY _____

REMINDER - PLEASE INITIAL
THIS DOCUMENT TO IND
THAT YOU HAVE RECE

DOCKETED FOR ___8/11/04___

Appeal Brief due

PTO-90C (Rev. 10/03)

Advisory Action	Application No.	Applicant(s)
	09/609,073	MARSHALL ET AL.
	Examiner	Art Unit	
	Steven Ashburn	3714	

The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 12 July 2004 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE. Therefore, further action by the applicant is required to avoid abandonment of this application. A proper reply to a final rejection under 37 CFR 1.113 may only be either: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114.

<u>PERIOD FOR REPLY</u> [check either a) or b)]

a) ☒ The period for reply expires <u>6</u> months from the mailing date of the final rejection.

b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection. ONLY CHECK THIS BOX WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

1. ☐ A Notice of Appeal was filed on _____. Appellant's Brief must be filed within the period set forth in 37 CFR 1.192(a), or any extension thereof (37 CFR 1.191(d)), to avoid dismissal of the appeal.

2. ☐ The proposed amendment(s) will not be entered because:

    (a) ☐ they raise new issues that would require further consideration and/or search (see NOTE below);

    (b) ☐ they raise the issue of new matter (see Note below);

    (c) ☐ they are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

    (d) ☐ they present additional claims without canceling a corresponding number of finally rejected claims.

        NOTE: _____.

3. ☐ Applicant's reply has overcome the following rejection(s): _____.

4. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

5. ☒ The a)☐ affidavit, b)☐ exhibit, or c)☒ request for reconsideration has been considered but does NOT place the application in condition for allowance because: *See Continuation Sheet*.

6. ☐ The affidavit or exhibit will NOT be considered because it is not directed SOLELY to issues which were newly raised by the Examiner in the final rejection.

7. ☒ For purposes of Appeal, the proposed amendment(s) a)☐ will not be entered or b)☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.

    The status of the claim(s) is (or will be) as follows:
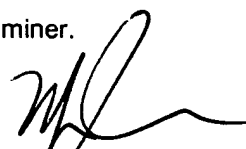
    Claim(s) allowed: _____.

    Claim(s) objected to: _____.

    Claim(s) rejected: *2-19 and 38-48*.

    Claim(s) withdrawn from consideration: _____.

8. ☐ The drawing correction filed on _____ is a)☐ approved or b)☐ disapproved by the Examiner.

9. ☐ Note the attached Information Disclosure Statement(s)(PTO-1449) Paper No(s). _____.

10. ☒ Other: *See Continuation Sheet*

MARK SAGER
PRIMARY EXAMINER

Continuation of 5. does NOT place the application in condition for allowance because: The examiner has fully considered the applicant's arguments but finds them unpersuasive. The rejections set forth in the office action dated Feb. 11, 2004 are respectfully maintained..

Continuation of 10. Other: The amended claims are rejected based on the same reasons set forth in the office action dated Feb. 11, 2004.

EVIDENCE APPENDIX G
COPY OF BRENNER ET AL., U.S. PATENT NO. 5,830,068

EVIDENCE APPENDIX H
<u>COPY OF LAWLER ET AL., U.S. PATENT NO. 5,805,763</u>

DOT/FAA/CT-96/1

FAA Technical Center

Atlantic City International Airport, NJ 08405

HUMAN FACTORS DESIGN GUIDE

For Acquisition of Commercial-Off-The-Shelf Subsystems,
Non-Developmental Items, and
Developmental Systems

Dan Wagner, ACT-530

Joseph A. Birt
Michael Snyder
System Resources Corporation (SRC)

James P. Duncanson
CTA INCORPORATED

January 15, 1996

Final Report and Guide

This document is available to the public through the
National Technical Information Service,
Springfield, Virginia 22161

US Department
of Transportation

Federal Aviation
Administration

Office of the Chief Scientific
and Technical Advisor for
Human Factors
AAR 100

1. Report No. DOT/FAA/CT-96/1	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle HUMAN FACTORS DESIGN GUIDE (HFDG) For Acquisition of Commercial Off-The-Shelf Subsystems, Non-Developmental Items, and Developmental Systems		5. Report Date January 15, 1996	
		6. Performing Organization Code ACT-500	
7. Author(s) Dan Wagner (ACT-530), Joseph A. Birt and Michael D. Snyder (SRC), and James P. Duncanson (CTA)		8. Performing Organization Report No. DOT/FAA/CT-96/1	
9. Performing Organization Name and Address U.S. Department of Transportation Federal Aviation Administration Technical Center Atlantic City International Airport, NJ 08405		10. Work Unit No (TRAILS)	
		11. Contract or Grant No. F2203C	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Headquarters Program Management & Planning Division 800 Independence Avenue, SW Washington, DC 20591		13. Type of Report and Period Covered Technical Report	
		14. Sponsoring Agency Code ALM-100	

15. **Supplementary Notes:** This work was accomplished under the direction of Dan Wagner, ACT-530, as part of the FAA's Airway Facilities Human Factors R,E & D Program. The Program is sponsored by Brenda Boone, ALM-100, and managed by Lawrence Cole, AAR-100. Technical consultation was provided by Kermit Grayson.

16. **Abstract**
The Human Factors Design Guide (HFDG) provides reference information to assist in the selection, analysis, design, development, and evaluation of new and modified Federal Aviation Administration (FAA) systems and equipment. A preliminary edition was a draft standard developed at the Human Factors Laboratory of the FAA Technical Center. This 1996 edition converts the preliminary draft document to a guide and incorporates expert comments that were collected in 1994 and 1995 from selected reviewers. It is primarily focused on FAA ground systems and equipment such as those that are managed and maintained by Airway Facilities. This guide covers a broad range of human factors topics that pertain to automation, maintenance, human interfaces, workplace design, documentation, system security, safety, the environment, and anthropometry. This document also includes extensive human-computer interface guidance.

17. Key Words Human factors, Human-equipment interfaces, Human-computer interfaces, maintenance automation, design for maintenance, workplace design, user documentation, system security, personnel safety, environment		18. Distribution Statement This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of pages 1012	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Foreword

The *Human Factors Design Guide (HFDG) for Acquisition of Commercial-off-the-Shelf (COTS) Subsystems, Non-Developmental Items (NDI), and Developmental Systems* is a comprehensive reference tool that will help human factors professionals within the Federal Aviation Administration (FAA) and contractor organizations to efficiently carry out FAA human factors policy.

A preliminary edition of the present document was a draft standard developed at the Human Factors Laboratory of the FAA Technical Center. This 1996 edition converts the preliminary draft document to a guide and incorporates expert comments that were collected in 1994 from selected reviewers.

FAA Order 9550.8, *Human Factors Policy*, states that:

> Human factors shall be systematically integrated into the planning and execution functions of all FAA elements and activities associated with system acquisition and system operations. FAA endeavors shall emphasize human factors considerations to enhance system performance and capitalize upon the relative strengths of people and machines. . .

The *Acquisition Strategy Paper* required by the new FAA *Acquisition Management System, April, 1997*, states that:

> . . . human factors will be considered during architectural and engineering design to achieve effective human performance during operations, maintenance, and support.

The HFDG was developed by the Aviation Simulation and Human Factors Division at the FAA Technical Center to consolidate and capitalize upon multiple sources of human factors design and evaluation guidelines. It provides FAA system modernization programs access to the most applicable human factors guidance. This guide is intended to overcome the imitations associated with using other design standards in an FAA environment.

Application of this design guide is not a substitute for in-depth professional human factors practice. The Acquisition Management System also refers to a military human factors process standard, MIL-STD-46855, which calls for planning human factors activities and procedures. Both human factors acquisition guidelines and processes are to be professionally applied. The use of the HFDG requires expert professional judgment on its application to new systems and equipment.

This document compiles extensive guidance from diverse and exhaustive sources for human factors applications integral to the procurement, acquisition, design, development, and testing of FAA systems, facilities, and equipment. It will aid in identifying

functional, product, and NAS specification requirements and in ensuring acceptable human factors practice and products.

This edition of the HFDG is applicable to COTS and NDI procurements as well as new developmental system or equipment acquisitions. The relationship between hardware and software subsystems and the human subsystem's characteristics must be determined and tested in advance of commitments to procure and implement COTS and NDI equipment and systems. These characteristics can include human roles, organizations, interfaces, tasks, training, and human performance effectiveness.

This version of the HFDG remains primarily focused upon FAA ground systems and equipment such as those that are managed and maintained by Airway Facilities. Although good human factors practices and principles apply to all FAA systems, this guide is not directed at special considerations in Air Traffic Control operations, aircraft maintenance, aircraft or airborne equipment certification, or FAA's regulatory certification for aviation personnel, although many of the HFDG provisions apply to those environments. Future editions will more directly address these areas of NAS development and operations.

The HFDG draws heavily from human factors information published by the Department of Defense, National Aeronautics and Space Administration, and Department of Energy. The FAA recognizes the excellent quality of information found in many of the technical documents and handbooks written by these agencies.

Request for feedback comments. Comments for corrections or improvements are welcome. Comments can be made at any time by using the form at the end of the document.

Contents

Section 8 contents

8.4 Data entry

8.5 Data display

**8.9.8
Accommodating
assistive devices**

8 Human-computer interfaces

This section contains criteria and guidelines governing human-computer interfaces. The topics covered include: (1) modes of human-computer interaction, (2) basic screen design, (3) windowing, (4) data entry, (5) data display, (6) user guidance, (7) data communication, (8) input devices, and (9) accommodating people with disabilities.

8.1 User-computer interaction

Interaction between a user and a computer is a two-way communication process; the user issues a command, and the computer responds. Two terms are commonly used to designate this process, "sequence control" and "interactive control." This HFDG uses the latter.

A series of commands and responses is called a "dialog." There are eight major types of dialogs: (1) question and answer, (2) language, (6) query language, (7) natural language, and (8) direct manipulation. The types are not exclusive, rather they are often used in combination, for example, windowing systems tend to use a combination of menu selection and direct manipulation.

> **Definitions.** A **transaction** is a user action paired with an associated computer response (or vice versa). It is the smallest unit of user-computer interaction. A **dialog** is a structured series of transactions.

8.1.1 General

□ **8.1.1.1 Consistent control actions.** Interactive control actions should be consistent in form, means, and consequence from one transaction to another, from one task to another, and from one application to another.

> **Discussion.** This guideline is extremely important for users of multiple applications. For example, if a user of a system being designed or selected must control several diverse operating systems or inconsistent control functions, then high error rates, extensive training, and low human reliability may be a consequence.

□ **8.1.1.2 System matched to user abilities.** Interactive control systems should be adaptable to individual differences and should accommodate the variety of user abilities expected, whether novice or expert. If applicable, systems and applications should provide relatively helpful or self-explanatory operations for novice or infrequent users, and relatively efficient operations for experienced users.

□ **8.1.1.3 User control.** The user-computer interaction should give users the feeling that they control the system, not that the system controls them.

■ **8.1.1.4 User control of pace.** The user, not the computer, shall control the pace of control entries by explicit actions.

■ **8.1.1.5 Explicit user control.** A user shall complete a control entry or action through an explicit action, such as pressing the **Enter** key; the system shall not interrupt the user's actions to indicate that it recognizes the command.

> **Exceptions.** Certain exceptions to this rule are given in the following instances. For example, expert users of a windowing system may prefer implicit control when assigning focus to a window. That is, their action of moving the cursor into a window automatically makes it the active window. Also, expert users entering very structured data may work more efficiently if the application supports auto-tabbing between fields.

■ **8.1.1.6 Simplicity.** Interactive control shall be simple, flexible, and adaptive, as well as consistent and compatible with the lowest anticipated user skill level. Interactive control shall be logical in terms of user task sequences and functions.

▢ **8.1.1.7 Minimal user actions.** Interactive control logic should permit completion of a task with the minimum number of actions, consistent with user abilities.

▢ **8.1.1.8 No repetitive entry of information.** If the same information is required for more than one transaction in the same application, the user should only supply it once; the application should supply it automatically thereafter.

▢ **8.1.1.9 User perspective.** A sequence of transactions should be designed to be logical from the perspective of the user, not from the perspective of computer processing or ease of programming.

■ **8.1.1.10 Transaction wording.** The wording used in a transaction shall be consistent with the user's frame of reference and with the wording used in user guidance.

▢ **8.1.1.11 User expectations.** The result of any correct control entry should be compatible with a user's expectations.

▢ **8.1.1.12 Minimal memory load.** The short-term memory requirements on users should be minimized by such means as making displays and interactive sequences self-evident and by providing on-line help and tutorials.

▢ **8.1.1.13 Customized interaction.** If practical, users should be able to customize the information displayed and the control options available to match their individual needs. This ability should not be provided if users share systems or applications in a way that would allow one user's customization to negatively affect or impact another user.

■ **8.1.1.14 Multiple users.** If a system or application can be used by more than one user at the same time, control actions by any user shall not interfere with those of any other user.

> **Exception.** An exception to this rule is when real-time group interaction, such as shared document editing, in which only one user at a time may interact with the document, and other users are unable to access it during that time.

■ **8.1.1.15 Paging and scrolling.** If a display shows only part of the data currently available for display, the system or application shall provide paging, scrolling, or both.

> **Definitions.** Scrolling is a method used to move through the contents of a window or list in a dialogue box using the scroll bar or scroll arrows. Paging is process of scrolling through data one page at a time.

□ **8.1.1.16 Upper-lower case equivalence.** In interpreting user-generated control entries, the system or application should treat upper and lower case letters as equivalent.

□ **8.1.1.17 Canceling or undoing actions.** User actions should be easy to cancel or undo (see paragraph 8.2.1.7).

□ **8.1.1.18 Names of control functions.** The names of interactive control functions should be semantically congruent with natural usage, especially for paired opposites. For example, if **Up** is the command to move the cursor up, the opposite command would be **Down**, not **Lower**.

□ **8.1.1.19 Closure.** The sequence of user actions and computer responses that accomplishes a task should be designed to give the user a sense of completion or closure at the end. Not only does this give the user a sense of accomplishment, but it lets him or her know when to go on to another task.

□ **8.1.1.20 Interactive paradigm.** Applications should base their interactions on an object-action paradigm, that is, a user first selects an object, then specifies an action.

> **Discussion.** The objects selected may be controls, text entities, or graphic entities. More than one object may be selected.

□ **8.1.1.21 User control.** Users should control the pace of the interaction with an application. The application should initiate an action only in response to an explicit user input. If appropriate, users should be able to specify when a process occurs, and they should be able to interrupt or terminate a process.

□ **8.1.1.22 Immediate feedback.** Users should receive an immediate, visible response to every action.

Discussion. If the result of the action is not immediate, some other visible response may be made. For example, a button selected might be highlighted, the pointer might change shape, or a message might be displayed.

□ **8.1.1.23 Visual cues.** An application should inform users when they can and when they cannot take an action by providing visual cues indicating when it can accept input, when it is momentarily unavailable, and when it is unavailable during extended processing.

□ **8.1.1.24 Prompts.** If a user must perform several actions to complete a task, the application should prompt the user with the actions that need to be performed.

□ **8.1.1.25 Ignoring user actions.** An application should ignore user actions (except for user interrupts) made during periods of time when input cannot be accepted. It should disable the pointing device and keyboard when input might have destructive effects. User inputs made during processing by the application should not be stored for execution after processing is completed. While users should not be able to override disabling, users should be able to stop a process if desired (see section 8.1.3.3).

□ **8.1.1.26 Error detection.** If a user attempts to initiate an invalid action, the application should display a message stating that the action is invalid and should not initiate the action. If the attempted action is part of a series of actions, the user should only have to correct the invalid action; he or she should not have to repeat the entire sequence of actions.

■ **8.1.1.27 Confirmed destruction.** Applications shall be designed so that users must confirm any destructive action that cannot otherwise be reversed or "undone."

8.1.2 System response time

In designing any application, response time is critical. The response of an application is dependent on hardware and other processes requiring central processor unit (CPU) use. For example, a multitasking system may be slowed by other concurrent applications and therefore, is hard to quantify. Thus, the rules in this section need to take into account such factors.

■ **8.1.2.1 Appropriate system response time.** The response time of a system to a user action shall be appropriate to the type of transaction, the time constraints of the task, and any specific data processing requirements. Responses to menu selections, function key presses, and most entries during graphic interaction shall appear to a user to be immediate. Other response times shall match the user's perception of the complexity of the transaction, with apparently simpler transactions having faster responses.

■ **8.1.2.2 Maximum system response times.** System response times shall not exceed the values given in exhibit 8.1.2.2 for the system tasks listed.

Exhibit 8.1.2.2 Maximum system response times for routine system tasks

System interpretation	Response time definition	Maximum response time (sec)
Key response	From key depression until positive response, for example, "click" or display echo	0.1
Key print (echo)	From key depression until appearance of character	0.2
Page turn	From end of request until first few lines are visible	1.0
Page scan	From end of request until text begins to scroll	0.5
Data field entry	From selection of field until visual verification	0.2
Function selection	From selection of command until response	2.0
Pointing	From input of point to display of point or pointing device	0.2
Drawing, sketching	From input of point to display of point, line, arc, etc.	0.2
Local update	Change to image or display using local data base, for example, new menu list display	0.5
Host update	Change where data are at host in a readily accessible form, for example, a display scale change	2.0
File update	Image or display update requiring access to a host file	10.0
Simple inquiry	From command until display of a common message	2.0
Complex inquiry	Response message that requires seldom used calculations in graphic form	10.0
Error feedback	From entry of input until error message appears	2.0

■ **8.1.2.3 Variability of system response time.** The variability of system response times for processing various types of control actions shall be minimized. For processing in the range of 0 to 2 sec, variability shall not exceed 5 percent; for processing in the range 2 to 5 sec, variability shall not exceed 10 percent; for processing longer than 5 sec, variability shall not exceed 15 percent.

■ **8.1.2.4 Acknowledgement of delayed processing.** If the processing of a control entry must be delayed because of computer processing of a prior entry or entries, the current control entry shall be acknowledged.

■ **8.1.2.5 Notification of display completion.** If the generation of a display is time consuming, and if it is not otherwise obvious, the system shall notify the user when the display is complete.

□ **8.1.2.6 Response-time-induced keyboard lockout.** If application processing prohibits acceptance of keyboard input and no keyboard buffer is available, the application should lockout the keyboard until the application can accept input.

> **Definition. Keyboard lockout** is a state determined by an application in which the application does not accept input from the keyboard.

■ **8.1.2.7 Lockout duration.** Temporary lockout of a keyboard or other device due to processing of a transaction control entry shall be minimized.

■ **8.1.2.8 Lockout indication.** If an application incorporates keyboard lockout, it shall provide a clear indication to users when the keyboard is locked out and when it is not. One way this might be done is to change the shape of the cursor or pointer to a watch or hourglass.

□ **8.1.2.9 Lockout override.** An application that incorporates keyboard lockout should also provide a means for overriding the lockout, for example, by assigning a function key to have this effect. If lockout override is provided and it is invoked, the system should not reset and lose any processing that was completed before the override was invoked.

8.1.3 System-initiated information

8.1.3.1 Prompting

■ **8.1.3.1.1 Prompting.** A system or application shall (1) prompt users for all required input parameters, (2) request additional or corrected information as needed, (3) provide orientation (as to the computer's processes to users) during transactions, and (4) indicate any errors that are detected.

■ **8.1.3.1.2 Prompt contents.** If the computer is waiting for input from a user, it shall indicate clearly where on the screen the input is expected and, to the extent possible, what information is expected.

■ **8.1.3.1.3 Location of prompts.** Prompting messages shall appear in a consistent location on the screen, for example, at the beginning of the next line to be typed, in the data field where an entry is to be made, at a command input line, or within a menu window from which a selection is to be made.

□ **8.1.3.1.4 Duration of prompts.** If a computer requests information from a user, any instructions about how to supply the

information should remain visible until the user complies or takes some other action.

❑ **8.1.3.1.5 User-selected level of prompting detail.** A system or application should permit users to select the level of detail they want in prompts. This capability should not be provided if the system or application is shared in a way that would allow one user's selection to affect another user.

8.1.3.2 Feedback

■ **8.1.3.2.1 Entry acknowledgement.** Every user action shall result in a response from the system. This response shall not exceed the maximum time intervals presented in Exhibit 8.2.16.2.

■ **8.1.3.2.2 Periodic feedback.** If the system takes more than 2 seconds to respond, it shall provide periodic feedback to the user indicating that normal operation is occurring (see paragraph 8.2.7.2).

❑ **8.1.3.2.3 Periodic feedback messages.** Successive periodic feedback messages should differ in wording from presentation to presentation, or be otherwise indicated. For example, three successive messages might be: (1) "Processing search -- please wait." (2) "Search continuing -- please wait." (3) "Processing search -- wait please."

■ **8.1.3.2.4 Completion of processing.** If the computer response to a user action is lengthy, the computer shall give a clear and positive indication when processing is complete.

8.1.3.3 System-initiated interrupts

■ **8.1.3.3.1 System interrupts.** A system or application shall interrupt a user only when necessary to prompt the user for a response, to provide essential feedback, or to inform the user of errors.

■ **8.1.3.3.2 "Working" indication.** If a system or application takes more than 2 seconds to complete an operation initiated by a user action, and during this time it is incapable of accepting further input from the user, it shall inform the user that action is continuing. For example, the system might display a "working" message, or a symbol such as a watch or an hourglass (see exhibit 8.8.3.6.1).

> **Discussion.** A dynamic aspect to the "working" message is highly desirable. For example, the message might display the percent of processing that has been completed or that remains, with the percentage updated regularly. If this is not possible, a display that changes with time is still desirable, for example, a row of dots, with a new dot added periodically.

8.1.3.4 Status information

□ **8.1.3.4.1 Availability of status information.** Information about the current status of the system should be available to users at all times. As appropriate to the system, this information should be provided automatically or upon user request.

> **Discussion.** System status information might include information about data processing status, system availability, operational mode, system load, other users, and external systems.

□ **8.1.3.4.2 Status of alarm settings.** Users should be able to obtain status information concerning current alarm settings, for example, the dimensions or variables covered and the values or categories established as critical.

□ **8.1.3.4.3 Status of other systems or users.** If interaction with other systems or users is required, status information about the other systems or users should be available.

8.1.3.5 Alarms

■ **8.1.3.5.1 Distinctive and consistent alarms.** Alarm signals and messages shall be distinctive and consistent for each class of event, for example, a signal alerting a user to an incoming message would be different from an signal alerting a user to a hazardous condition.

■ **8.1.3.5.2 Acknowledging and terminating alarms.** A system or application shall provide users a means of acknowledging critical and noncritical alarms, and of turning off alarm signals once the alarms have been acknowledged or the condition generating the alarm has been corrected. Procedures for acknowledgment and termination shall not decrease the speed and accuracy of operator reaction to the alerting situation.

■ **8.1.3.5.3 Feedback about alarms and alerts.** Users shall be provided informative feedback for actions that trigger alarms and alerting signals. If necessary, users shall be able to request help and related information for the operation and processing of critical and noncritical alarms, messages, and signals.

■ **8.1.3.5.4 Special acknowledgement of critical alarms.** If a user must acknowledge a special or critical alarm in a unique way, for example, with a special combination of key strokes, this special acknowledgement shall not inhibit or slow the response to the condition initiating the alarm.

■ **8.1.3.5.5 Alarm reset.** A system or application shall provide users with a simple means for turning off an auditory alarm without erasing any displayed message that accompanies the auditory signal.

□ **8.1.3.5.6 User setting of alarm parameters.** If appropriate to the task, a system or application should allow a user to set the parameter or condition that results in a software-generated alarm, alert, or status message. Some examples of parameters or conditions are priorities, percentages, and absolute values or ranges of values. User setting of parameters should not be allowed if: (1) the settings by one user might affect the reception of alarms by another user, (2) the settings might affect the safety of systems, equipment, or personnel, or (3) alarm parameters are determined by functional, procedural, or legal requirements.

8.1.3.6 Routine messages

■ **8.1.3.6.1 Routine feedback.** The system shall provide users consistent, routine feedback regarding such activities as control entries, computer processing, and print requests.

□ **8.1.3.6.2 User control.** If appropriate, users should be able to specify the level or type of system message they want to receive.

□ **8.1.3.6.3 Clarity of purpose.** The wording of routine messages should make clear to the user that they provide status or feedback information, not that they indicate errors or requests for a user action.

8.1.4 User-initiated interrupts

■ **8.1.4.1 User interruption of transactions.** A system or application shall permit a user to interrupt or terminate the current transaction. Each type of interrupt shall have a separate control option and a distinct name. The following types of interrupts may be provided: **Cancel, Escape, Backup, Restart, Abort, Stop, Pause-Continue,** and **Suspend**.

■ **8.1.4.2 Stored or entered data.** User interruptions shall not change or remove stored or entered data, with the exception of the Cancel interrupt.

■ **8.1.4.3 Backup (or Go-back).** A nondestructive **Backup** or **Go-back** option shall be provided to return the display to the last previous transaction.

■ **8.1.4.4 Cancel (or Undo).** If appropriate, a system or application shall provide a Cancel or Undo option that will erase changes just made by a user and restore the current display to its previous state.

■ **8.1.4.5 End, Exit, or Stop.** If appropriate, a system or application shall provide an **End, Exit,** or **Stop** option to conclude a repetitive transaction sequence.

■ **8.1.4.6 Pause and Continue.** If appropriate, a system or application shall provide **Pause** and **Continue** options that will

interrupt and later resume, respectively, a transaction sequence without any change to data entries or control logic for the interrupted transaction.

■ **8.1.4.7 Indicating Pause status.** If a **Pause** option is provided and selected, the system or application shall provide an indication that the transaction sequence has been halted. The system or application shall prompt the user to select **Continue** to resume the interrupted sequence.

■ **8.1.4.8 Restart (or Revert).** If appropriate, a system or application shall provide a **Restart** or **Revert** option that will cancel entries made in a defined transaction sequence and will return the user to the beginning of the sequence. If a restart will result in the loss of data or changes, the system shall require a confirming action by the user.

■ **8.1.4.9 Review.** If appropriate, a system or application shall provide a nondestructive **Review** option that will return to the first display in a defined transaction sequence, permitting the user to review a sequence of entries and make necessary changes.

■ **8.1.4.10 Suspend.** If appropriate, a system or application shall provide a **Suspend** option that permits a user to preserve the current state of a transaction while leaving the system and to resume the transaction at a later time.

■ **8.1.4.11 Indicating Suspended status.** If a system or application provides a **Suspend** option, it shall display an indication that a transaction has been suspended whenever the option has been selected. The system shall prompt the user with information on how to resume the suspended transaction at his or her next log on. For example, the user might see: "Type **Exit** to return to application."

8.1.5 Error management

8.1.5.1 General

□ **8.1.5.1.1 User-detected errors.** A user should be able to stop a control process at any point in a sequence to correct an error.

■ **8.1.5.1.2 Appropriate response to all entries.** A system or application shall provide an appropriate response to all possible control entries, correct and incorrect. For example, the selection of an incorrect function key might result in a message listing the appropriate selections.

□ **8.1.5.1.3 System detection of error type.** A system or application should be able to distinguish among program errors, equipment failures, and operator errors and, if a failure results in a shutdown, allow for minimum loss of work performed.

□ **8.1.5.1.4 Fast error detection.** User errors should be detected and reported by the system as soon as possible, so that they are easier to correct.

■ **8.1.5.1.5 Immediate data correction.** If a user has completed a data entry transaction and an error is detected, the user shall be able to make corrections directly and immediately.

□ **8.1.5.1.6 Prompting command correction.** If a system or application does not recognize an element of a command entry, the system should prompt the user to correct that element rather than require re-entry of the entire command.

□ **8.1.5.1.7 Display duration.** Notices, alerts, and informational displays should remain visible to a user until he or she responds with an appropriate action.

■ **8.1.5.1.8 Enter action for corrections.** A system or application shall require an explicit user action to re-enter corrected material after a user has completed correcting an error. The enter action for re-entry shall be the same as the enter action for the original entry.

■ **8.1.5.1.9 Return to main interaction.** A system or application shall provide an easy means to return to the main dialog after error correction.

■ **8.1.5.1.10 User confirmation of destructive actions.** If a control entry (including log off) will result in a change in stored data, procedures, or system operation, particularly if it is not easily reversible, the system or application shall notify the user and require a confirmation of the action before implementing the action. The notification shall explicitly warn the user of potential loss of data. The **Enter** key shall not be used for the confirmation action.

■ **8.1.5.1.11 Flexible "go back" for error correction.** A system or application shall allow a user to go back easily to previous steps in a transaction sequence in order to correct an error or make any other desired change.

■ **8.1.5.1.12 Undo control action.** A system or application shall provide an Undo operation that immediately reverses the last previous control action.

■ **8.1.5.1.13 Error recovery.** All conditions and information relevant for user recovery from an error shall be displayed to the user. Users shall be able to correct the error immediately. Error messages and error feedback about the data or control entry shall be given within 2 seconds of the time the error is detected.

8.1.5.2 Error messages

□ **8.1.5.2.1 System-detected need for help.** To the extent practicable, a system or application should detect inappropriate user entries and actions, automatically interrupt the task, and

either suggest an appropriate entry or action or ask the user to confirm or clarify his or her intentions.

> **Examples.** The system might provide a message when it detects an error, an out-of-range response, a missing parameter, a duplicated entry, or an unusually long pause on the part of the user.

■ **8.1.5.2.2 Confirmation messages.** If a user entry might cause the loss or destruction of data or a disruption of a system, the system shall display a cautionary message and require that the user confirm the entry. This includes logging off the system (see also paragraph 8.3.10.2.10).

■ **8.1.5.2.3 Multilevel messages.** If appropriate, the system shall provide more than one level of error messages, with successive levels providing increasingly detailed levels of explanation.

□ **8.1.5.2.4 Multiple errors.** If a system detects multiple errors, it should describe the first error and inform the user of the total number of additional errors. The cursor should be moved to the location of the first error. If appropriate, the system should provide a means for the user to request sequential display of the additional error messages.

□ **8.1.5.2.5 Nondisruptive error messages.** The display of error messages should not disrupt ongoing user activity, for example, an error message should not be displayed until a user has completed an entry. In general, error messages should be displayed within 4 seconds after the user completes the entry in which the error is detected.

■ **8.1.5.2.6 Coding of warning messages.** Messages that require special user attention shall be coded appropriately and distinctively (see section 8.5.4).

□ **8.1.5.2.7 Content of error messages.** If applicable, error messages should state the error detected, the input field containing the error, and the corrective action.

■ **8.1.5.2.8 Wording of error messages.** Error messages shall be brief, specific, and task-oriented (see also section 10.2.3).

□ **8.1.5.2.9 Tone of error messages.** In general, error messages should be worded as advice or suggestions.

□ **8.1.5.2.10 Correcting errors.** If possible, after detecting an error, the system should prompt users to reenter only the portion of the entry or command that is in error. That is, users should not have to reenter the entire entry.

■ **8.1.5.2.11 Cursor placement.** After an error message is displayed, the cursor shall be placed at the location of the entry in error.

- **8.1.5.2.12 Instructions and error messages.** Instructions and error messages shall appear in a consistent location on the screen (same as 8.2.1.2.4).

8.1.5.3 Command interaction errors

- **8.1.5.3.1 Command editing.** A system or application shall permit a user to edit an extended command during its composition, that is before taking an explicit Enter action.

- **8.1.5.3.2 Command correction prompting.** A system or application shall prompt a user to correct an element of a command entry that is not recognized or that is logically inappropriate. Whenever possible, the faulty command shall be retained in the command entry area of the display, with the cursor automatically positioned at the incorrect item, and with an advisory message displayed that describes the problem. The user shall not have to re-enter the entire line or command. The message shall make clear to the user what corrective action is required.

- **8.1.5.3.3 Unrecognized commands.** If a menu selection, function key, or command entry is invalid or inoperative at the time of selection, (for example, if a user attempts to print a document while in an editing mode), no action should result except the display of an advisory message. This message should tell the user what is wrong and which functions, options, or commands are appropriate.

- **8.1.5.3.4 Errors in stacked commands.** If an error is detected in a series of stacked command entries, the system shall operate consistently in one of the following modes: (1) execute commands up to the point of error, or (2) require the user to correct any errors before executing any of the commands.

- **8.1.5.3.5 Partial execution of stacked commands.** If only a portion of a stack of commands can be executed, the system or application shall notify the user and provide appropriate guidance to permit correction, completion, or cancellation of the inexecutable command.

- **8.1.5.3.6 Stacked command execution.** If the system detects an error in a stack of commands it is processing, it shall notify the user and promptly (within 4 sec) provide guidance to permit correction, completion, or cancellation of the stacked commands.

8.1.6 Transaction and control options

- **8.1.6.1 User-specified transaction timing.** If appropriate to task requirements, users shall be able to specify transaction timing. For example, users might be able to specify when a transaction starts, when it is completed, and the periodic scheduling of repeated transactions.

■ **8.1.6.2 User memory load.** The number of mnemonics, codes, special or long sequences, and special instructions that users may need to learn shall be minimized.

■ **8.1.6.3 Prompting control entries.** The system or application shall provide the user whatever information is required to guide control entries.

> **Examples.** Prompts may be incorporated into a display at any point in a transaction sequence that will be helpful, or prompts may appear in response to a request for help. The selected prompts must be used consistently.

■ **8.1.6.4 List of basic control options.** A list of basic control options that are always available to a user shall be easily displayable. This list can serve as a "home base" or starting point for control entries. An example is the system-level menu.

□ **8.1.6.5 Appropriate specific options.** A list of the control options that are specifically appropriate for a particular transaction should be displayed in the working display or by user command.

□ **8.1.6.6 Option wording.** The wording of control options should be task oriented, reflecting a user's view of the current transaction, for example, if users use the term "assign," the control option control option should also be **Assign**.

□ **8.1.6.7 Option presentation.** The options presented in a list of basic options should be grouped, labeled, and ordered according to their: (1) logical function, (2) sequence, (3) frequency, or (4) criticality of use. If these ordering schemes are in conflict, default to the higher level order.

■ **8.1.6.8 Option code display.** If users must select options by entering codes, the code associated with each option shall be displayed in a consistent manner and shall be distinct from other codes. If possible, the codes shall be intuitive.

■ **8.1.6.9 Displaying control defaults.** If control is accomplished by keyed command or option code entries and a default entry is defined, the default shall be displayed to the user.

■ **8.1.6.10 Initial cursor position for pointing devices.** If a user must select among displayed options using a pointing device, the cursor shall be placed on the default option when the display appears (same as paragraphs 8.1.11.7.2 and 8.4.1.6.4).

> **Definition.** A **cursor** is a marker on the display screen that indicates the position where the computer expects the next input or will display the next output. The cursor may be positioned under computer control or by the user.

■ **8.1.6.11 Initial cursor position for keyboards.** If a user must select among displayed options using a keyboard, the cursor shall be placed on the default option in the control entry area (with that

control entry area having implicit input focus) when the display appears (see "discussion" in paragraph 8.3.4.3.1) (same as paragraphs 8.1.11.7.3 and 8.4.1.6.5).

■ **8.1.6.12 Consistent Continue option.** At any step in a defined sequence of transactions, if there is only a single appropriate next step, the system or application shall provide a consistent control option. For example, if data entry is involved, an explicit **Enter** or Tab control option signalling entry shall be used rather than a **Continue** or **Next** action.

□ **8.1.6.13 Options at completion of a transaction.** A transaction should never leave a user without further available options and should provide next steps or alternatives, for example, **Continue, Abort,** or **Go to main directory.**

□ **8.1.6.14 Command stacking.** A system or application should permit, but not require, a user to enter a sequence (or"stack") of command names, abbreviations, and option codes as a single stacked command. For example, a stack of commands might execute a complete task. Stacked commands must be entered in the same order that would be used if they were entered singly. If there is an error in a stack, the system or application should highlight the point of error and prompt the user for a correct entry.

■ **8.1.6.15 Punctuation of stacked commands.** Required punctuation of stacked commands shall be minimized. A delimiter to separate commands shall be adopted and used consistently. For example, the slash (/) might be adopted as the delimiter, and a stacked command might be: **Sort/Save/Transmit.** If possible, the delimiter shall be as intuitive as possible by using an ampersand (&), a "plus" sign (+), or a comma (,).

□ **8.1.6.16 User-defined stacks (macros).** A system or application should allow a user to define a series of graphical- or character-based control entries, assign the series a name (macro), and subsequently enter the series by simply entering that name.

8.1.7 Abbreviations

■ **8.1.7.1 Abbreviations.** If a system or application uses abbreviations in its user-computer interface, the abbreviations shall be unique, distinct, and unambiguous. Their use shall not confuse users and shall not add to system operation time.

> **Definition.** An **abbreviation** is any shortened form or abridgment of a word, expression, or phrase used to conserve space or time. Thus, the term abbreviation includes initializations and acronyms.

■ **8.1.7.2 Use of abbreviations.** The use of abbreviations shall be minimized. If an abbreviation must be used for a term, the

abbreviation shall be selected or constructed by the first of the following methods that applies:

a.   by selecting the abbreviation from FAA Order 7340.1,

b.   by selecting the abbreviation from the U.S. Government Printing Office *Style Manual*,

c.   by constructing an abbreviation following the rules in the U.S. Government Printing Office *Style Manual*.

■ **8.1.7.3 Definitions of abbreviations.** If a system or application uses abbreviations in its user-computer interactions, it shall provide an easy on-line, context-sensitive means for a user to learn the definition of an abbreviation.

■ **8.1.7.4 New abbreviations.** If new abbreviations are needed, they shall be developed according to the rules of the U.S. Government Printing Office *Style Manual*.

8.1.8 Interaction method

■ **8.1.8.1 Selection of interaction type.** The interaction type shall be selected to be appropriate to the task requirements, the characteristics of the system, and the abilities of the users. The appropriateness of the major types of interaction for these requirements, characteristics, and abilities are listed here and summarized in exhibit 8.1.8.1.

a.   The **question and answer** interaction type is appropriate if:

    (1)   the task is routine data entry,

    (2)   the characteristics of the data are known and ordering can be constrained,

    (3)   users are expected to have little or no training, and

    (4)   computer response is expected to be moderately fast.

b.   The **form filling** interaction type is appropriate if:

    (1)   flexibility in data entry is needed,

    (2)   users are expected to be moderately trained,

    (3)   computer response may be slow, and

    (4)   an aid in composing complex control entries would be helpful.

Exhibit 8.1.8.1 Appropriateness of interaction types for various task requirements, system characteristics, and user abilities

Task, system characteristic user ability	Question and answer	Form filling	Menu selection	Function keys	Command language	Query language	Constrained natural language
Arbitrary control or data entry sequences					X		
Poorly defined or broad interface definition							X
Unpredictable information retrieval						X	X
Wide range of control entries					X		
Frequent transactions				X			
Small or constrained command choice set			X	X			
Complex control		X					
Large command set			X				
Routine data entry	X						
Entry order constrained	X						
Data entry flexibility needed		X					
Little arbitrary data input			X				
Slow computer response time		X					
Fast computer response time	X		X				
High training of users					X		
Moderate training of users		X				X	
Little or no training of users	X		X				X

c.   The **menu selection** interaction type is appropriate if:

(1)   tasks involve choices from constrained sets of alternatives,

(2)   little entry of arbitrary data is required,

(3)   users are expected to have little training,

(4)   a command set is too large for users to remember, and

(5)   computer response is relatively fast.

d.   The **function key** interaction type is appropriate for use in conjunction with other types if:

    (1)   tasks require only a limited number of control entries, or

    (2)   an immediate means for accomplishing frequent control entries or transactions is desirable.

e.   The **command language** interaction type is appropriate if:

    (1)   tasks involve a wide range of control entries,

    (2)   users are expected to be highly trained or will use the system frequently, and

    (3)   control entries may be mixed with data entries in arbitrary sequence.

f.   The **query language** interaction type is appropriate if:

    (1)   tasks emphasize unpredictable information retrieval, and

    (2)   users are expected to be highly trained.

g.   The **constrained natural language** interaction type is appropriate if:

    (1)   task requirements are wide ranging or poorly defined, and

    (2)   users are expected to have moderate training.

h.   The **direct manipulation** interaction type is appropriate when tasks mimic physical manipulation of concrete objects, such as positioning graphical objects, moving blocks of text, and resizing objects. It is also appropriate for casual system users and users expected to have little or no training.

■ **8.1.8.2 Distinctive display of control information.** Displays shall be designed so that features such as prompts and messages relevant to the interactive method are distinctive in position and format.

■ **8.1.8.3 Hierarchical levels.** If hierarchical levels are used to control a process or sequence, the number of levels shall be minimized. Display and input formats shall be similar within levels, and the system shall indicate the current position within a sequence (see also paragraph 8.1.11.3.4).

8.1.9 Question-answer

■ **8.1.9.1 Singular presentation of questions.** Questions shall be presented one at a time and shall not require the user to answer more than one question at a time. To the extent possible, users shall be provided a default or a list of the most appropriate responses from which they may select the desired response.

□ **8.1.9.2 Display of interrelated answers.** If a system poses a series of questions to the user, and the answer to the current question is dependent upon how a previous question was answered, answers to all questions within the series should be displayed until all questions have been answered.

■ **8.1.9.3 Sequence compatibility with source document.** If questions require entry of data from a source document, the question sequence shall match the data sequence within the source document.

8.1.10 Form-filling

■ **8.1.10.1 Consistency.** The forms and formats of form-filling interactions shall be consistent and logical throughout an application and related applications.

■ **8.1.10.2 Default entries.** Wherever possible, default entries shall appear in their fields when a form is displayed in form-filling interactions.

■ **8.1.10.3 Default listing.** A default listing or screen shall be provided in which authorized users may view and change default settings of fields.

■ **8.1.10.4 Other applicable sections.** In addition to the criteria and guidelines in this section, those of section 8.4, in particular section 8.4.2, shall also apply to form-filling interactions.

8.1.11 Menus and menu selection

The use of menus as an interaction method is widespread, often in conjunction with other methods, direct manipulation, in particular. Menus are usable with little or no training on the part of the user. If the meanings of the options are clear, the user can be guided step-by-step through an application. Menus do have some disadvantages, however; they can slow down an experienced user; they can occupy a considerable amount of display space; and, in complex sequences, users may become lost in the menu structure.

Definitions. A **menu** is a list of options from which a user makes a selection or selections. An **option** is one of the selectable items in a menu. **Selection** is the action a user makes in choosing a menu option. Selection may be accomplished by pointing, by typing, or by a pressing a function key.

8.1.11.1 General

■ **8.1.11.1.1 Menu titles.** A menu shall describe or explain the options listed under it. The title shall be easily distinguished from the options.

■ **8.1.11.1.2 Consistent style.** Menus throughout an application shall conform to a single style of interface, for example, OSF/Motif™, Open Look™, Microsoft Windows™, or Macintosh™ (same as paragraph 8.4.1.1.1).

■ **8.1.11.1.3 Consistent wording and ordering.** Menus and options that appear in different displays and contexts shall be consistent in wording and ordering (same as paragraph 8.4.1.1.2).

■ **8.1.11.1.4 Consistent with command language.** If menu selection is used in conjunction with command language interaction, the wording of menu options shall be consistent with the command language.

> **Definition. A command language** is a limited programming language used strictly for executing a series of commands.

□ **8.1.11.1.5 Response time and display rate vs. menu length.** The design of menus should take into account the response time and display rate of the system. If the computer response time to a user action is long, menus should have relatively more options; if display rate is slow (that is, if it takes a long time to complete the drawing of a display), menus should have relatively fewer items.

> **Discussion.** If the computer's response time is long, then menus should be broad and shallow and if the display rate is slow, menus should be narrow and deep.

□ **8.1.11.1.6 Number of options.** The number of options in a menu should not be more than ten or less than three (same as paragraph 8.3.7.3.6).

■ **8.1.11.1.7 Display of all options.** A menu shall display explicitly and completely all options available to a user at the current step in a transaction sequence.

■ **8.1.11.1.8 Distinguishing unavailable options.** If a menu contains options that are temporarily unavailable, the unavailable options shall be displayed but clearly distinguishable from available options. For example, unavailable options might be displayed at reduced intensity ("grayed out") (same as paragraphs 8.1.11.2.7, 8.4.1.1.6, and 8.3.7.3.7).

■ **8.1.11.1.9 Distinguishing types of options.** If a menu contains options of different types, for example, options that lead to other menus and options that are values that can be entered in fields, the types shall be distinguishable. For example, options that lead

to other menus might be followed by a triangle that points to where the subsequent menu will appear (▷ or ▽). A menu option that requires additional information from the user might be followed by an ellipsis (...) (same as 8.4.1.1.5).

■ **8.1.11.1.10 Instructions.** Instructions pertaining to menus shall appear in a help window and in a consistent location on the display (same as paragraph 8.4.1.1.7).

▫ **8.1.11.1.11 Shortcuts for experienced users.** Experienced users should have a way to bypass the menu structure for frequently accessed options (see also paragraph 8.1.11.3.13).

■ **8.1.11.1.12 Stacking menu selections.** If the selection of options from menus is accomplished by entering codes, and if a series of selections can be anticipated before the menus themselves are displayed, users shall be able to combine those selections into a single, stacked entry that is equivalent to the series of selections, but without having the menus displayed.

> **Definition. Stacking** is the stringing together of commands so that they can all be executed with a single command.

■ **8.1.11.1.13 Menus distinct from other displayed information.** Menus that appear in displays that also contain other objects or information shall be distinct from the other objects or information on the screen (same as paragraph 8.4.1.1.8).

8.1.11.2 Menu formatting

▫ **8.1.11.2.1 Presentation of options.** With the exception of a menu bar, the options in a menu should be presented in a single vertical column, aligned and left-justified.

■ **8.1.11.2.2 Consistent menus and options.** If the same menu or option appears in different displays within an application, it shall be consistent in wording and ordering (same as paragraph 8.4.1.5.2).

■ **8.1.11.2.3 Logical grouping of menu options.** If applicable, the options in a menu shall be presented in logical groups (same as paragraph 8.4.1.5.3).

■ **8.1.11.2.4 Ordering groups of options.** Groups of options in a menu shall be ordered logically. If there is no apparent logical ordering, the groups shall be ordered by their expected frequency of use (same as paragraph 8.4.1.5.4).

■ **8.1.11.2.5 Ordering options within a menu or group.** If a group of options or a menu contains a small number of options, the options shall be ordered by importance, logical sequence, or frequency of use. If a group or menu contains a very large number of options, the options shall be ordered alphabetically (same as paragraph 8.4.1.5.5).

□ **8.1.11.2.6 Numbering menu options.** If task order is important, menu options should be numbered.

■ **8.1.11.2.7 Distinguishing unavailable options.** If a menu contains options that are temporarily unavailable, the unavailable options shall be displayed but clearly distinguishable from available options. For example, unavailable options might be displayed at reduced intensity ("grayed out") (same as paragraphs 8.1.11.1.8, 8.4.1.1.6, and 8.3.7.3.7).

8.1.11.3 Hierarchical menus

Large or complex menus can be presented as hierarchical menus.

> **Definition.** A **hierarchical menu** is a large menu that is organized as a multi-level, branching structure of smaller menus in which an option in a higher level menu is the name of another menu at the next lower level. The options in the lowest level menus are such things as commands or values; they are not the names of other menus.

□ **8.1.11.3.1 When to use.** Hierarchical menus should be used if there are many options (more than 10), and the options can be organized in a branching structure.

□ **8.1.11.3.2 Organizing and labeling hierarchical menus.** Hierarchical menus should be organized and labeled to guide the user within the hierarchical structure.

> **Example.** When a user selects an option from a menu, the menu and the selected option remain on display with the selected option highlighted, and the lower-level menu that results from the selection is displayed adjacent to the selected option.

■ **8.1.11.3.3 Consistent design and use.** The design and use of hierarchical menus shall be consistent across tasks and transactions within an application.

□ **8.1.11.3.4 Minimum number of levels.** A hierarchical menu structure should minimize the number of selections required to reach the desired option. This implies the use of broad, shallow structures as opposed to narrow, deep ones.

□ **8.1.11.3.5 Easy selection of important options.** Hierarchical menus should permit immediate user access to critical or frequently selected options.

■ **8.1.11.3.6 Indicating current position in menu structure.** An indication of the user's current position in a hierarchical menu structure shall be provided.

□ **8.1.11.3.7 Hierarchical menus in graphical user interfaces.** Hierarchical menu design in a graphical user interface should be as simple as possible; complex graphical structures should be avoided.

■ **8.1.11.3.8 Top level menu.** The top level menu in a hierarchical menu structure shall serve as a consistent starting point for control entries. A user shall be able to return easily to the top level at any time.

■ **8.1.11.3.9 Organization of a system-level menu.** The options of a system-level menu shall be grouped, labeled, and ordered in terms of their logical function, frequency of use, and criticality.

■ **8.1.11.3.10 Return to system-level menu.** A user shall be able to return to a system-level menu from anywhere in a hierarchical menu structure with one simple control action.

■ **8.1.11.3.11 Return to next higher level.** A user shall be able to return to the next higher level menu from anywhere in a hierarchical menu structure with one simple control action.

□ **8.1.11.3.12 Lower level menus.** The options contained in a menu below the top level should be logically related to each other.

□ **8.1.11.3.13 Bypassing menu selections.** The system or application should allow a user to bypass a series of menu selections by making an equivalent command entry (see also paragraph 8.1.11.1.11).

□ **8.1.11.3.14 Software navigation aids.** Software navigation aids should be provided to assist the user in quickly selecting the desired menu (for example, a tree diagram or organization chart). The aid should permit a user to select a menu directly, without going through intermediate steps.

8.1.11.4 Menu bars

This section presents criteria and guidelines for menu bars.

Definition. A **menu bar** is a menu that is usually displayed horizontally across the top of a display screen. The options on a menu bar are usually the names of other menus.

□ **8.1.11.4.1 When to use.** A menu bar should only be used if the display screen size and resolution permit fast and accurate movement of the cursor onto the options.

□ **8.1.11.4.2 Visibility of menu bar options.** Menu bar options should remain visible at all times.

8.1.11.5 Pull-down menus

This section presents criteria and guidelines for pull-down menus.

Definition. A **pull-down menu** is a menu that appears when a menu bar option is selected.

□ **8.1.11.5.1 When to use.** Pull-down menus should be used rather than pop-up menus if the position of the cursor on the screen is not important for information or option retrieval (same as 8.4.1.3.1).

> **Discussion.** The advantage of pull-down menus over pop-up menus is that pull-down menus always have a visual cue in the form of a menu bar.

■ **8.1.11.5.2 Consistent location.** Pull-down menus shall always appear immediately below the option whose selection leads to their appearance (same as paragraph 8.4.1.3.2).

8.1.11.6 Pop-up menus

Pop-up menus can be very useful in data entry. They can present to a user the permissible entries for a field, thus (1) eliminating the need for the user to remember the entries, (2) preventing invalid entries, and (3) eliminating potential typing errors.

> **Definition.** A **pop-up menu** is a menu that is associated with a particular object on a display, for example, a pop-up menu listing acceptable command options close to the immediate work area. This is particularly useful for large displays, where the work site may be relatively removed for the menu bar.

□ **8.1.11.6.1 When to use.** Pop-up menus should be used only if it is critical to the application that users be able to access functions without moving the pointing device. They should not be the only method for accessing operations, since the operations are hidden from view, requiring users to remember where they are and how to access them (same as paragraph 8.4.1.4.1).

■ **8.1.11.6.2 Pop-up menu location.** A pop-up menu shall appear in a location that is coordinated with the location of the pointer (same as paragraph 8.4.1.4.2).

■ **8.1.11.6.3 Selecting an option using a pointing device.** A user shall be able to select an option on a pop-up menu by moving the pointer onto the desired option and clicking the appropriate button (same as paragraph 8.4.1.4.3).

> **Explanation.** This method is preferred to holding the button down while moving the cursor and releasing it to make a selection. The deliberate click method is less prone to error.

8.1.11.7 Selecting options

■ **8.1.11.7.1 Equivalence of input devices.** The system or application shall provide a user the ability to use any of the input devices available to select a menu option. For example, if a user has both a pointing device and a keyboard available, he or she shall be able to use either to select an option (same as paragraph 8.4.1.6.1).

■ **8.1.11.7.2 Initial cursor position for pointing devices.** If a user must select among displayed options using a pointing device, the cursor shall be placed on the default option when the display appears (same as paragraphs 8.1.6.10 and 8.4.1.6.4).

■ **8.1.11.7.3 Initial cursor position for keyboards.** If a user must select among displayed options using a keyboard, the cursor shall be placed on the default option in the control entry area (with that control entry area having implicit input focus) when the display appears (see "discussion" in paragraph 8.3.4.3.1) (same as paragraphs 8.1.6.11 and 8.4.1.6.5).

■ **8.1.11.7.4 Feedback for menu selection.** If no computer response is immediately observable when a user selects an option, the software shall provide some other acknowledgment of the selection. For example, the software might display a watch, hourglass, or a message stating the delay remaining or the elapsed time (same as paragraph 8.4.1.6.6).

□ **8.1.11.7.5 Abbreviated entries.** If menu selection is by code entry, the application should accept both the complete and minimum distinguishing abbreviated forms of the code. For example, an application might accept **Q, QU,** and **QUIT** as equivalent.

□ **8.1.11.7.6 Menu selection by pointing.** If menu selection is the primary interactive method, and especially if selections are made from extensive lists of options, selection by pointing device should be provided (same as paragraph 8.4.1.6.2).

■ **8.1.11.7.7 Size of selectable area.** The effective pointing area for menu options shall be as large as is consistently possible. The area shall be at least the displayed option label plus a half-character distance around that label.

■ **8.1.11.7.8 Two-action activation.** If menu selection is accomplished with a pointing device, activation shall consist of two actions: (1) designation, in which a user positions the cursor on the desired option (with that option being highlighted when the pointer is on the menu option), and (2) activation, in which a user makes a separate, explicit control entry (clicking the appropriate mouse button) (see the "discussion" in paragraph 8.8.3.7.2).

□ **8.1.11.7.9 Number of selections per menu.** A user should be allowed to select only one option from a menu. If the menu is divided into groups, a user should be able to select only one option from each group, although users may be able to select multiple files from a menu (same as paragraph 8.4.1.6.7).

8.1.11.8 Titles and wording of options

■ **8.1.11.8.1 Wording of options.** The wording of options shall use familiar terminology (such as those used in industry), but shall distinguish each option from every other option in the menu.

□ **8.1.11.8.2 Options as commands.** Options should be worded as commands to the computer, not questions to the user.

□ **8.1.11.8.3 Titles for groups of options.** If the options in a menu are grouped and titled, the titles should be comprehensible and unique.

■ **8.1.11.8.4 Appearance of group titles.** The titles of groups of options shall appear in a format that is clearly distinguishable from that of the options themselves.

8.1.11.9 Coding options Mnemonic coding can be of help to users. Mnemonic letters are the easiest codes to remember; numbers are intermediate; and nonmnemonic letters are the most difficult. Letters as codes also have a numerical advantage over numbers (there are 26 letters as opposed to only 10 numerals). Numerals have the advantages of making sequencing clear, being easier to locate on a keyboard by nontypists, and allowing a user to know immediately how many options there are.

□ **8.1.11.9.1 Conveyed meaning of coding.** To the extent possible, an option code should suggest the meaning of the option it represents. An example would be the use of the first letter or letters of each option as the code.

■ **8.1.11.9.2 Consistent coding.** The coding of menu options shall be consistent throughout an application and related applications.

□ **8.1.11.9.3 Letter vs. numeric codes.** Letter and numeric codes should not be used in the same menu.

■ **8.1.11.9.4 Numeric coding.** If menu options are numbered, numbering shall start with 1, not with 0.

■ **8.1.11.9.5 Displaying option codes.** If menu options are coded, the codes shall be displayed with their options in a consistent, distinctive manner.

> **Examples.** If numeric coding is used, the numerals might appear immediately to the left of the options. If mnemonic coding is used, the mnemonic letter or letters might be emboldened (**Undo**) or underlined (Undo).

8.1.12 Function keys This section contains criteria and guidelines for assigning functions to function keys and using them (see also section 8.8.2).

□ **8.1.12.1 Single function.** If feasible, a function key should be assigned only one function.

■ **8.1.12.2 Consistency within an application.** If an application includes different operational modes and function key assignments that vary from mode to mode, to the extent possible, equivalent or similar functions shall be assigned to the same function key for all modes.

■ **8.1.12.3 Consistency across applications.** If the same function exists in related applications, it shall be assigned to the same key in all applications.

■ **8.1.12.4 Feedback.** Feedback shall be provided for function key activation. If the activation does not result in an immediately observable response from the computer, the user shall be given some other form of acknowledgment or feedback. No system function shall be activated without an indication to the user. If system functioning results in a long delay, the user shall be given feedback periodically.

□ **8.1.12.5 "Soft" function keys.** If "soft" function keys are used, representations of the function keys should be presented on the screen. These representations should have the same spatial configuration as the "hard" function keys, and they should be located as near as possible to the hard keys, probably at the bottom of the screen (assuming the keyboard is normally positioned directly below the screen).

> **Definition.** A **soft key** is an area on the screen that represents a function key. If a function key is assigned more than one function in an application, an associated soft key can be labeled with the function that is currently assigned to the key.

□ **8.1.12.6 "Soft" function key activation.** If a screen includes "soft" function keys, and if the application provides a pointing device, a user should be able to initiate a function both by pressing the corresponding "hard" function key and by selecting the soft key with the pointing device.

■ **8.1.12.7 Disabling of unused function keys.** Function keys that are unassigned or that are assigned a function that is not applicable at the moment shall be "disabled" by the computer. When some function keys are active and some are not, the system shall indicate which are active.

> **Discussion.** This might be done by displaying only the active keys as "soft" keys on the screen, or by displaying active "soft" keys differently from inactive ones.

■ **8.1.12.8 Easy return to base-level functions.** If functions assigned to a set of keys change as a result of user selection, it shall be easy for the user to return them to the initial, base-level functions. One way this might be done is to include the equivalent of a "Main Menu" key in all sets other than the base set of function keys.

□ **8.1.12.9 User-defined functions (macros).** If appropriate, users should be able to define their own functions and assign them to function keys, either temporarily or permanently. This capability should not be provided if macros defined by one user might be used inadvertently by another user.

■ **8.1.12.10 Single key operation for continuously-available functions.** If a function is available continuously, it shall be initiated by simply pressing its assigned function key or selecting a corresponding "soft" key.

■ **8.1.12.11 Frequently-used functions.** If a function will be used frequently, if its use is critical, or if its timely use is critical, it shall be initiated with a single key operation.

■ **8.1.12.12 Single key press.** A function key shall perform its labeled function with a single press of the function key. Function keys shall not change function with repeated key presses without an indication of the new function or change in mode.

□ **8.1.12.13 Relationship of functions assigned to the same key.** If two or more functions are assigned to the same function key and they are accessed by simultaneously pressing the function key and another key, such as **Shift, Ctrl,** or **Alt,** the functions should be logically related to each other.

□ **8.1.12.14 Relationship of sets of functions.** If two or more sets of functions are assigned to function keys and they are accessed by simultaneously pressing a function key and another key, such as **Shift, Ctrl,** or **Alt,** the logical relation between the sets should be consistent from one key to another.

> **Example.** In a text processing application, one set of functions might apply to lines, another to paragraphs, and another to pages.

■ **8.1.12.15 Labeling single-function keys.** A function key assigned a single function shall have a label on the keycap that clearly identifies the function and clearly distinguishes that function from others.

■ **8.1.12.16 Labeling multifunction keys.** If a key is used for more than one function, the user shall be informed which function is currently available. One way to accomplish this is to display a label on a "soft" key on an adjacent portion of the screen.

■ **8.1.12.17 Indicating status.** If applicable, the active or inactive status of a function key shall be indicated. One way to accomplish this is to change the appearance of displayed labels on the screen, for example, dimming inactive keys or displaying one status in dark text on a light background and the reverse for the other state.

□ **8.1.12.18 Labeling of menu items selectable with function keys.** If items from a menu are to be selected using function keys, the items should be labeled with function key numbers (that is, **F1, F2,** and so on) and if screen "real estate" is not at a premium, they should appear as "soft" key labels above the function keys.

■ **8.1.12.19 Importance and frequency of use.** Functions shall be assigned to keys in accordance with their importance and frequency of use. For example, an emergency function might be given the most prominent position, or the most frequently used function might be given the most convenient location.

■ **8.1.12.20 Safeguarding.** Function keys that have potentially disruptive consequences shall be safeguarded. Safeguarding may take the form of physical protection, software disabling, interlocks, or multiple key combinations.

8.1.13 Command language design

■ **8.1.13.1 Functional command language.** A command language shall be designed so that users can enter commands in terms of functions desired, without concern for internal computer processing, storage, and retrieval mechanisms.

> **Definition.** A **command language** is a limited programming language used strictly for executing a series of commands.

■ **8.1.13.2 Consistent syntax.** Command language syntax shall be consistent within an application and across related applications.

> **Definition.** The **syntax** of a command language is the set of rules governing the language, for example, rules about the order in which parts of a command occur, or rules about punctuation in commands.

□ **8.1.13.3 Complexity of command language.** The complexity of a command language (its syntax) should be minimized, especially if there will be many untrained or infrequent users.

■ **8.1.13.4 Layered command language.** The command language shall be designed so that its features (functions) are organized in groups for ease of learning and use.

□ **8.1.13.5 Command stacking.** Users should be able to make control entries in accordance with task requirements, entering more than one command before entering an "execute" command, if that best meets the task requirements.

■ **8.1.13.6 Command entry area.** Each display shall provide a command entry area that is located consistently from display to display, for example, at the bottom of the screen.

■ **8.1.13.7 Distinctive wording of commands.** Words in a command language shall be distinctive from one another, emphasizing significant differences in function.

■ **8.1.13.8 Consistent wording of commands.** All words and their abbreviations in the command language shall be consistent in meaning and spelling from one transaction to another and from one task to another.

■ **8.1.13.9 Familiar wording.** Words for use in command language dialog shall be chosen to reflect the user's point of view and shall correspond to the user's operational language.

□ **8.1.13.10 Abbreviation of commands.** If a command language is necessary for the system, and if the operators may be power users, then commands should also have abbreviated forms having five or less characters.

□ **8.1.13.11 Selection of commands.** Commands should be designed to aid memory.

□ **8.1.13.12 Alternate wording.** If a system will have many novice or infrequent users, it should recognize a variety of synonyms or alternative syntax for each word defined in the command language.

□ **8.1.13.13 "Word" length.** The length of an individual input word, such as a command or a key word, should not exceed seven characters.

■ **8.1.13.14 Characters.** Commands shall have at least one alphabetic or numeric character. Commands consisting of only nonalphanumeric characters (such as $ or @) shall not be used.

■ **8.1.13.15 Punctuation.** The use of punctuation in commands shall be minimized. If a delimiter is needed, one delimiter, such as the slash (/), shall be used throughout an application and related applications.

□ **8.1.13.16 Blank spaces.** Blank spaces should not be required or interpreted by an application.

■ **8.1.13.17 Spelling errors.** Commands shall be selected so that likely spelling errors do not result in valid commands (for example, using DEL for Delete and SEL for Select might result in this sort of error, since the D and S keys are adjacent on QWERTY keyboards).

■ **8.1.13.18 Editing commands.** Users shall be able to edit textual commands after they are typed, but before they are executed, using standard editing techniques (see section 8.4.4.2).

■ **8.1.13.19 Execution.** Once a textual command has been composed, an explicit enter or execute action by the user shall be required.

■ **8.1.13.20 Confirmation of a command.** If the execution of a command might result in a delay, the deletion or modification of data, or other potentially adverse consequences, the system or application shall inform the user of the nature of the consequence and request that the user confirm the command unless an **UNDO** command is available.

■ **8.1.13.21 Unrecognized commands.** If the system or application does not recognize a command a user has entered, the system or application shall inform the user and request the user to revise or replace the command.

8.1.14 Query and natural language

This section contains criteria and guidelines for database queries.

> **Definitions.** A **data base** is a set of interrelated data stored in a computer. A **query** is the process of specifying, locating, and retrieving data matching specified characteristics from a data base.

8.1.14.1 General

□ **8.1.14.1.1 Ease of use.** A query language should be easy to learn and use.

□ **8.1.14.1.2 Interactive.** A query language should permit on-line, interactive use as opposed to batch or off-line use.

■ **8.1.14.1.3 Natural organization of data.** A query language shall be designed so that it considers the structure or organization of the data as perceived by the user group.

■ **8.1.14.1.4 Task-oriented queries.** In composing a query, a user shall be able to simply specify which data are requested, without having to tell the system how to find the data.

□ **8.1.14.1.5 User assistance.** A query language should assist users in the construction of complex queries and in narrowing down overly broad queries.

■ **8.1.14.1.6 Large-scale retrieval confirmation.** If a query will result in a large or time-consuming data retrieval, the system or application shall notify the user of the amount of data or time and request that the user confirm the transaction or take further action to narrow the query before proceeding. The user shall be able to interrupt the retrieval process (see section 8.1.4).

□ **8.1.14.1.7 Logical combination queries.** A query language should permit the use of logical combinations in the formation of a query. Combinations that might be permitted include "and," "or," and "not."

□ **8.1.14.1.8 Subsequent queries.** A query language should permit the linking of sequential queries so that subsequent queries can be based on the results of prior queries. An example might be: "Of those records retrieved, how many...."

■ **8.1.14.1.9 Flexible queries.** If natural language query is permitted, the system or application shall allow users to employ alternative forms when initiating queries.

> **Example.** A system might accept all of the following as equivalent:
>
> Update network display within three miles.
> Update network display in a three mile radius.
> Update network display out to three miles.

□ **8.1.14.1.10 Error detection and correction.** A query language should detect and notify users of syntax errors in queries and assist them in correcting the errors.

□ **8.1.14.1.11 Formats matched to user needs.** Query and display formats should be matched to the nature of the searches users will make. If appropriate, more than one format should be provided.

□ **8.1.14.1.12 User preferences.** To the extent practicable, users should be able to choose the type of format (pictorial, verbal, or tabular) they prefer for queries and displays.

8.1.14.2 Query screen design

■ **8.1.14.2.1 Applicable criteria and guidelines.** Query screen design shall conform to the criteria and guidelines in sections 8.3, 8.4, and 8.5.

□ **8.1.14.2.2 Relevant information only.** Query screens should include only information that is relevant to the task, that is, information necessary to perform actions, make decisions, or answer questions.

□ **8.1.14.2.3 Frequently-used information.** The most frequently used information should be located in the upper left portion of a screen and, if multiple screens are involved, on the first screen or screens.

8.1.14.3 User requirements

□ **8.1.14.3.1 Importance of search terms.** A query language should permit users to rank order the search terms in importance and use this ranking in displaying the retrieved information.

□ **8.1.14.3.2 Redisplay.** A query language should retain the results of the previous search so that they can be redisplayed without repeating the search.

□ **8.1.14.3.3 Spelling and word variants.** A query language should recognize:

a. spelling variations, for example gray and grey,

b. acronyms,

c. inverted word order, for example, television monitor and monitor, television, and

d. truncations.

□ **8.1.14.3.4 Punctuation.** A query language should automatically remove or ignore punctuation in search terms.

□ **8.1.14.3.5 Word roots.** A query language should include a means for reducing words to their root forms, for example, by removing suffixes, and searching for the roots.

□ **8.1.14.3.6 Exceptions.** A query language should provide for a list of exceptional words that are accepted literally, that is, that are not reduced to their roots.

□ **8.1.14.3.7 Appearance of output.** The appearance, print format, and organization of the output should be natural and acceptable to the users. Users should be able to specify report formats.

□ **8.1.14.3.8 Assisting the user.** A query language should assist users in formulating searches to ensure maximum usefulness of the search results.

8.1.14.4 Usability

□ **8.1.14.4.1 Commands.** If used, commands should be in an easy-to-learn, user-oriented system language. They should be clear, unambiguous, and distinctive.

□ **8.1.14.4.2 Minimal user effort.** The number of keystrokes required of users should be minimized.

■ **8.1.14.4.3 Messages.** Messages to the user shall conform to the criteria and guidelines in section 8.1.3.

□ **8.1.14.4.4 Ease-of-use features.** A query language should provide the following features to make it easier to use:

     a.   reuse of frequent queries,

     b.   user definition of macros,

     c.   keyboard accelerators,

     d.   automatic periodic backup,

     e.   a **Restore** utility to recover backup data, and

     f.   a **Pause** and **Resume** capability that would allow a user to stop working with the query language and resume at a later time.

8.1.14.5 Searching

□ **8.1.14.5.1 Searching operations.** A query language should provide the following searching operations to users:

     a.   a **Select** operation that enables users to select the desired data base,

     b.   **Create** and **Erase** operations that enable users to create and erase data sets,

c. **Combine** operation that enables users to combine data sets,

d. a **Report** operation that enables users to format, name, specify, display, print, and save a query,

e. a **Restrict** operation that enables users to restrict the output of a retrieval set,

f. a **Save** operation that enables users to save the results of a search,

g. a **Search history** operation that enables users to view a list of previous search commands upon request.

❑ **8.1.14.5.2 Control operations.** A query language should provide users the following control operations:

a. a **Mark** operation that stores the current field value for future reference, for example, marking a field or record for deletion,

b. a **Describe** operation that enables users to receive a detailed explanation or description of the current field value,

c. a **Drop** operation that drops the current field from the structure, and

d. a **Status** operation that enables users to request status information.

❑ **8.1.14.5.3 Query formulation operations.** A query language should provide the following query formulation operations:

a. a **Select** operation that identifies the fields from tables and functions that will appear in the query results,

b. a **Compile** operation that generates and validates an executable operation,

c. a **Run** or **Do query** operation that causes execution of the query,

d. a **Show** operation that allows various presentations of a tabular result and that could be used to present a preview of the results of a query or report,

e. a **Modify** operation that allows users to make changes in the definition of an existing query or report, and

f. a **Save** operation that allows storage and repeated use or modification of a query.

❑ **8.1.14.5.4 Abbreviations.** If abbreviations are used, they should be significantly shorter than the unabbreviated terms. Truncation

is the preferred form of abbreviation. A query language should recognize both the abbreviated and the unabbreviated term.

□ **8.1.14.5.5 Search time feedback.** A query language should inform users if a search will take more than a short time to complete or will overload the computer, and it should prompt the user to confirm, modify, or terminate the search.

□ **8.1.14.5.6 Additional operations.** A query language should provide the following additional operations:

a. a **Browse** operation that enables users to navigate through a data base,

b. a **Report format** operation that enables users to format the results of queries as reports,

c. a **Search index** operation that enables users to view the list of words and phrases available for searching, including a link to a data base thesaurus to suggest additional search terms,

d. a **Proximity searching** operation that enables users to search for words or terms in a positional relationship with word index fields, for example, titles or abstracts,

e. a logical search operation using the logical operators and, or, and not,

f. an iterative operation that enables users to define a search, view the results, refine the search, view the results, and so on,

g. an operation to specify a range of values for searching,

h. an operation to specify fields for searching,

i. an operation to specify field values for searching,

j. an operation to order field values, for example, numerically or alphabetically, and

k. an operation to search across files that enables users to obtain the number of references including the search term in all potential data bases.

8.1.14.6 Multiple levels

□ **8.1.14.6.1 Accommodating users differing in experience.** A query language should accommodate users with different levels of experience.

□ **8.1.14.6.2 Changing levels.** Users should be able to change the level at which they interact with the language at any time during a session.

□ **8.1.14.6.3 Context-sensitive help.** Context-sensitive help should be available upon user request at all levels.

□ **8.1.14.6.4 Novice level.** At the novice level, a query language should enable a user to begin work with little or no training.

> **Discussion.** A novice interface may contain only a subset of the search capabilities and fewer searchable fields, with the result that it may not attain the same specificity or variety of search techniques.

□ **8.1.14.6.5 Prompting novices.** At the novice level, a query language should prompt users to select options from lists and should provide explanations of the options.

□ **8.1.14.6.6 Commands for novices.** The command set for novices should be fewer and simpler than the command set for experts.

□ **8.1.14.6.7 Commands for experts.** A query language for experts should allow the expert users to enter more than one command at a time.

8.1.15 Graphical controls

Icons may be used to represent operations, processes, and data structures graphically, and they may be used as a means of exercising control over system functions, components, and data structures.

8.1.15.1 Icons

■ **8.1.15.1.1 Resolution.** Iconic representation shall not be used if display resolution is low.

■ **8.1.15.1.2 Description.** An icon shall consist of a graphic image and where space permits, an identifying label. To the extent possible, the image shall represent or suggest the application or document it represents.

■ **8.1.15.1.3 Labels.** Labels shall be the same as the title of the window, if possible, and it shall appear below the image. If the title is too long to fit, it shall be truncated, but shall be displayed in full when the icon has input focus (see paragraph 8.3.4.3.1).

> **Discussion.** If there are so many icons displayed that the labels become too small to read, users must be able to choose whether or not to display the labels.

■ **8.1.15.1.4 Consistency.** Icons shall be consistent within an application and across related applications.

□ **8.1.15.1.5 Uniqueness of icons.** Any window that can be iconized should have a unique icon that serves as a visual representation of the window.

□ **8.1.15.1.6 Icon design.** To the extent possible, icons should be simple line drawings that suggest the object or operation they

represent. Icons should be based on physical objects where possible. Humorous representations should be avoided.

> **Examples.** Icons may be designed to represent a process or operation literally (for example, a drawing of an aircraft), functionally (for example, a figure representing a network), or operationally (for example, a drawing of a pen in hand on paper).

☐ **8.1.15.1.7 Manipulation of icons.** If direct manipulation interaction is used, the system or application should use a pointing device as the primary means of manipulation, not a keyboard.

▪ **8.1.15.1.8 Icon menu.** An icon shall have an icon menu that contains the same options as its window's system menu with the exceptions of **Resize** and **Iconize**. If the window menu includes these options, they shall appear on the icon menu as unavailable. (Current versions of Apple Macintosh^{TM} do not support icon menus.)

▪ **8.1.15.1.9 Using the icon menu.** A user shall be able to display an icon menu by moving the pointer onto the icon and clicking the appropriate button. A user shall select an icon menu item using standard option selection methods (see section 8.4.1.6). A user shall be able to remove an icon menu by moving the pointer off the menu and clicking the appropriate button.(Current versions of Apple Macintosh™ do not support icon menus.)

▪ **8.1.15.1.10 Restoring the window.** A user shall be able to restore a window and any secondary windows that were displayed when the window was iconized by: (1) moving the pointer onto the icon and double-clicking the appropriate button, or (2) displaying the icon menu and selecting **Restore**.

▪ **8.1.15.1.11 Location of icons.** Unless specified otherwise by the application, icons shall be placed in the lower left corner of the screen, and arrayed in the order in which they are created, in rows from left to right and from bottom to top.

☐ **8.1.15.1.12 User preferences.** Users should have the option of changing the default location of icons. User-selected locations for icons should be retained across sessions.

☐ **8.1.15.1.13 Moving icons.** Users should be able to move icons using similar methods available for moving windows (see paragraphs 8.3.5.3 and 8.3.5.4).

8.1.15.2 Push buttons

☐ **8.1.15.2.1 Consistent appearance.** All push buttons in a window should have the same size and shape. The size should accommodate the largest label.

▪ **8.1.15.2.2 Labels.** A push button shall have a label. The label may be either text or graphic.

■ **8.1.15.2.3 Consistent labels.** Push button labels shall be consistent throughout an application and related applications.

□ **8.1.15.2.4 Text labels.** Push button labels should be short and unambiguous. The label should describe the results of pressing the button and reflect the action that will be taken by the application rather than the user.

□ **8.1.15.2.5 "Standard" actions.** To the extent possible, the labels of push buttons should be selected from the list of "standard" actions given in Appendix D.

■ **8.1.15.2.6 Activating a push button.** A user shall be able to activate a push button by moving the pointer onto the button and pressing the appropriate pointer button. The push button shall be highlighted while the pointer button is depressed. The control shall be activated when the pointer button is released, and the push button shall revert to its normal appearance. A user shall also be able to activate a push button using the keyboard.

■ **8.1.15.2.7 Default push buttons.** Default push buttons shall be clearly distinguishable from the other push buttons. For example, they may have an extra border, as illustrated in exhibit 8.1.15.2.7, by highlighting, or making them appear three-dimensional. A push button assigned an action that is potentially destructive shall not be designated as the default button.

Exhibit 8.1.15.2.7 Example of a default push button

| OK | Cancel | Help |
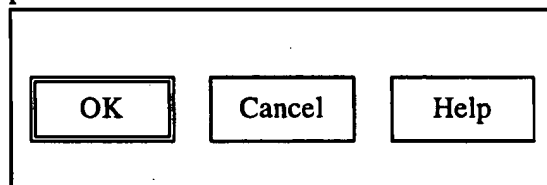
8.1.15.3 Radio buttons

Radio buttons (also known as exclusive buttons) are single, two-state choices, which are mutually exclusive from each other. For example, only one radio button can be "on" at a time. A radio button that is turned "on" will cause all of the other radio buttons to be turned "off."

■ **8.1.15.3.1 When to use.** Radio buttons shall be used if it is required that one and only one of a set of mutually exclusive options be selected. Exhibit 8.1.15.3.1 illustrates two possible types of radio button sets.

■ **8.1.15.3.2 Selecting an radio button.** A user shall be able to select a radio button using a pointing device by moving the pointer onto the radio button and clicking the appropriate device button. A user shall be able to select an radio button using the keyboard by moving a location cursor to the desired button (for example, using the arrow keys) and pressing the **Enter** key.

■ **8.1.15.3.3 Selected button highlighted.** One and only one button in a set of radio buttons shall be highlighted. If a user selects an unhighlighted button, that button shall be highlighted, and the

previously highlighted button shall have highlighting removed. Selecting a button that is already highlighted shall not change its state.

Exhibit 8.1.15.3.1 Two types of radio button sets

8.1.15.4 Check boxes

Check boxes (also known as nonexclusive buttons) are single, two-state choices. For example, a check box can be "on," (checked) or "off," not checked. A check box group is a collection of two-state choices, all of which apply to the same selected object. A check box that is turned "on" will not change the status of any other choice in the group.

> **Radar Type**
>
> O ARSR
>
> O ASR
>
> ● ASDE

> **Select a Site**
>
> O BAL
>
> O DCA
>
> ● ELM
>
> O PHL

■ **8.1.15.4.1 When to use.** Check boxes shall be provided if a user must be able to select any number, including none, of a set of options. For example, in specifying the appearance of text, a user might want to select both Bold and Italic.

■ **8.1.15.4.2 Selecting check boxes.** Check boxes shall have two states, selected and unselected. Users shall be able to toggle the states using both a pointing device and the keyboard. The state of each option shall be indicated.

> **Discussion.** One way the status might be indicated is by preceding each option with a "check box" that indicates whether or not the option is selected.

8.1.15.5 Sliders (scales)

Sliders are appropriate when users must set a value within a fixed range but the precise value is less important than relative position, for example, setting the volume level of a tone signal. Sliders are also appropriate for continuous, rather than discrete, variables.

> **Definition.** A **slider** is a control used to set a value and give a visual indication of the setting.

■ **8.1.15.5.1 Components of a slider.** A slider shall have a movable marker that indicates the current setting and a line or rectangular area along which it moves.

> **Discussion.** Tick marks and numeric values may be added to the line or rectangular area of the slider.

□ **8.1.15.5.2 Readout.** If appropriate, the slider should provide a numerical readout of the current setting.

■   **8.1.15.5.3 Slider operation.** Users shall be able to change the
    setting of a slider by moving the pointer onto the marker and
    dragging it.

■   **8.1.15.5.4 Labeling sliders.** A slider shall have a label or title
    that indicates the purpose of the slider, and, if appropriate, labels for
    the endpoints.

8.2 Basic screen design and operation

Screen design refers to the way information is arranged and
presented on a display screen. Different systems and applications
can perform a great variety of tasks. Some systems rely heavily
on data bases and do not require immediate user response to
information displayed on their screens. Other systems, such as
control systems, require that the users make immediate decisions
and issue commands based on information displayed to them.
The designer needs to understand the primary function of the
system being developed to provide an effective screen design.

8.2.1 Principles, features, and functions

8.2.1.1 General principles

□   **8.2.1.1.1 Simplicity.** Information should be presented simply and
    in a well-organized manner. Ways to achieve simplicity include
    the following:

    a.   The screen should appear to be orderly and clutter-free.

    b.   Information should be presented in consistent, predictable
            locations.

    c.   The language used should be plain and simple.

    d.   The means for moving around the screen and to related
            screens should be simple.

    e.   Interrelationships should be indicated clearly.

□   **8.2.1.1.2 Logical grouping.** Data items on a screen should be
    grouped on the basis of some logical principle.

□   **8.2.1.1.3 Minimal movement.** Screens should be designed to
    minimize both eye movement and pointer movement.

□   **8.2.1.1.4 What information to display.** The information to be
    displayed should be prioritized so that the most important or
    critical information can be displayed all the time, and less
    important or critical information can be displayed upon a user's
    request.

□   **8.2.1.1.5 Minimal information density.** The information density
    (the amount of information per unit area) of a screen should be

minimized by presenting only information that is essential to a user at any given time.

▫ **8.2.1.1.6 Screen density.** For text displays, screen density (the ratio of characters to blank spaces) should not exceed 60 percent; that is, not more than 60 percent of the available character spaces should be filled.

▫ **8.2.1.1.7 Integrated information.** If a user needs a variety of data to complete a task, those data should be provided in an integrated display, not partitioned in separate windows.

▪ **8.2.1.1.8 Directly usable form.** Information shall be presented to a user in directly usable form; a user shall not have to decode or interpret data.

8.2.1.2 Consistency

▪ **8.2.1.2.1 Consistent screen structure.** Screens throughout a system or application shall have a consistent structure that is evident to users.

▪ **8.2.1.2.2 Consistent screen elements.** Elements of screens such as headers, fields, and labels shall have the same appearance and relative location throughout a system or application.

▪ **8.2.1.2.3 Input prompts.** If applicable, an input prompt shall have a consistent location on all displays throughout a system or application.

▪ **8.2.1.2.4 Instructions and error messages.** Instructions and error messages shall appear in a consistent location on the screen (same as 8.1.5.2.12).

8.2.1.3 Context

▫ **8.2.1.3.1 Maintaining context.** An application should provide a means for ensuring that a user maintains an understanding of the context in which a task is being performed. For example, the application might display the results of those previous transactions that affect the current one, or it might display currently available options.

▪ **8.2.1.3.2 Highlighting.** When a user is performing an operation on a selected object in a display, that object shall be highlighted.

> **Discussion.** In many applications, at least two different methods of "selection" highlighting can be provided. The first of these highlighting methods occurs when the pointer comes to rest for a predetermined time on a "selected" object. This is sometimes referred to as "dwell emphasis" and it tells the user which object the computer "perceives" the user is about to select. This highlighting is normally "dim" white. The second type of highlighting occurs when an actual selection has been made and is normally a "bright" white.

■ **8.2.1.3.3 Display of context information.** Information intended to provide a context for the current user-computer interaction shall be distinctive in location and format, and shall be displayed consistently for all transactions within an application and among related applications.

■ **8.2.1.3.4 Distinctive position and format.** Displayed options, command entry areas, prompts, advisory messages, and other displayed items (such as titles and time signals) relevant to transaction control shall be distinctive in location and format.

■ **8.2.1.3.5 Operational mode.** If an application provides different operational modes, the current mode shall be continuously indicated to a user.

□ **8.2.1.3.6 Current context indication.** If the consequences of a control entry will differ depending upon the context established by a prior action, a continuous indication of current context should be displayed.

■ **8.2.1.3.7 Context-dependent actions.** The interpretation of a user action by the system shall be appropriate to the current context as determined by prior actions. A user shall not have to re-enter data he or she already entered in the current application session (see paragraph 8.1.1.8).

■ **8.2.1.3.8 Action history.** If appropriate, an application shall maintain a summary of the transactions that produced the current context and display it at a user's request. If desirable, an application shall link an **UNDO** feature to each step in the action history.

■ **8.2.1.3.9 Control parameters display.** A user shall be able to review all active control parameters upon request.

> **Discussion.** Control parameters can include current and default settings, as well as settings applicable to a particular mode of operation. These parameters apply to the application software and to parameters of an external system being remotely monitored and controlled.

8.2.1.4 Format

■ **8.2.1.4.1 Title.** Every screen shall have a title or header at the top. The title or header shall describe briefly the contents or purpose of the screen. The title or header shall be separate and distinguishable from the body of the screen. (Current versions of Apple Macintosh^{TM} do not offer titles on every screen.)

■ **8.2.1.4.2 Other reserved areas.** Any interactive elements used in a screen, such as prompts, menu bars, command lines, and message areas, shall appear consistently in the same screen location throughout the system or application.

■ **8.2.1.4.3 Layout of screen elements.** The layout of screen elements shall follow some organizing principle that users can recognize and apply.

□ **8.2.1.4.4 Minimal visual competition.** Information on a display screen should be organized so that visual competition among distinct items of information is minimized. For example, underlining words interferes with reading.

□ **8.2.1.4.5 Arrangement of screen elements.** Screens should be arranged so that there is a clear differentiation between instructions and data.

□ **8.2.1.4.6 Location of displayed instructions.** If instructions to users are included in a display, instructions on how to do something on the screen should precede (be located above or to the left of) the relevant object; instructions about the disposition of the completed screen should be at the bottom of the screen.

□ **8.2.1.4.7 Use of contrasting features.** Contrasting features such as inverse video or color should be used to call attention to critical screen components and urgent items.

■ **8.2.1.4.8 Abbreviations.** The use of abbreviations shall be minimized. If they are used, they shall be used consistently, and a key or built-in reference table shall be provided. Abbreviations shall not be followed by periods.

8.2.1.5 Displaying text

Continuous text can be broken up by the use of blank lines or by using lines drawn between or around portions of text. The use of different intensity levels is another possibility, but may be undesirable, depending upon the levels available and the ambient lighting conditions. This section contains criteria and guidelines for displaying text; the display of data is covered in section 8.5.

□ **8.2.1.5.1 Breaking up large blocks of text.** Large blocks of text should be broken into smaller, meaningful portions to minimize the amount of information requiring the user's attention at any given time.

> **Discussion.** The readability of large amounts of text may be improved by presenting the text in two columns.

□ **8.2.1.5.2 Lists.** A series or list of text elements should be presented vertically, not horizontally.

■ **8.2.1.5.3 Order of information.** If displayed information is to be used in some spatial or temporal arrangement, its order in the screen shall preserve that arrangement. If ordering by sequence, function, frequency, or importance is not appropriate, some other method, such as alphabetical or chronological, shall be followed.

■ **8.2.1.5.4 Primary viewing area.** Information that is particularly important or that requires immediate user response shall be displayed in the user's primary viewing area (see paragraph 7.2.1.6.8).

8.2.1.6 Scrolling and paging

■ **8.2.1.6.1 Stationary text.** Text information shall be stationary on the screen; it shall not be scrolled continuously except with user action.

■ **8.2.1.6.2 Paging.** If a screen contains too much data to display in a single frame, the data shall be partitioned into separately displayable pages.

□ **8.2.1.6.3 Labeling pages.** In a multipage display, each page should be labeled with the number of the page and the total number of pages, for example, "Page 2 of 3."

8.2.1.7 Initial display

■ **8.2.1.7.1 Initial display.** The initial display a user sees shall be a display that provides access to the highest level functions, resources, and applications available to the user. This includes access to the log on screen, user preference settings, utilities (such as a calculator, clock, and calendar), and system-level help.

> **Exception.** If a system is dedicated to a single application, the initial display may be the initial display of the application.

■ **8.2.1.7.2 Starting point.** In any display, it shall be obvious where the user is intended to "start." Ordinarily, this will be at the upper left part of the screen.

> **Discussion.** This might be accomplished by placing the pointer, if there is one, at that point, or by highlighting the "first" part of the screen. Another alternative is to highlight the last active window or icon.

8.2.1.8 Matching controls to users

□ **8.2.1.8.1 Minimizing the user's short-term memory load.** Windows should be designed to minimize the short-term memory load placed on a user as he or she performs the task called for by the window. A window should contain all relevant information and should allow a user to complete the task without having to refer to additional information.

□ **8.2.1.8.2 Selecting a mutually exclusive option.** Exclusive button sets (radio buttons) should be used when users need to choose one option from a small number (up to six) of mutually exclusive options. A menu should be used for up to 10 options, and a scrolling menu should be used for more than 10 options.

□ **8.2.1.8.3 Selecting nonmutually exclusive options.** If users need to select one or more of up to 10 nonmutually exclusive options, a nonexclusive button set (check boxes) should be used. For a larger set of options, a fixed or scrolling menu should be used.

❑ **8.2.1.8.4 Menus.** Menus should be used for selecting values and choosing from a set of related options.

❑ **8.2.1.8.5 Pop-up menus.** Pop-up menus should be used only if it is critical to the application that users be able to access functions without moving the pointing device. They should not be the only method for accessing operations, since the operations are hidden from view, requiring users to remember where they are and how to access them (same as paragraphs 8.1.11.6.1 and 8.4.1.4.1).

8.2.1.9 Arranging information to match user actions

❑ **8.2.1.9.1 Matching window layout to task.** Application designers should determine how users will activate the actions called for by a window and then design the window layout so that users can manipulate objects in ways that support task performance. For example, if an application generates information that will be presented a page at a time, the application should provide users with controls for performing paging operations. In addition, the objects in a window should be arranged so that users can move quickly and easily among them.

❑ **8.2.1.9.2 Matching window layout to users' "Natural" patterns.** Window layout should conform to users' natural scanning order and probable selection sequences. Usually, the order will be from left to right and top to bottom. For example, in button sets and menus, the most frequent choice should appear in the leftmost or top position.

❑ **8.2.1.9.3 Minimal user effort.** The amount of pointer movement and the number of keystrokes required to complete a task should be minimized.

8.2.1.10 Arranging information by importance

❑ **8.2.1.10.1 Location by importance.** The most important information and controls associated with a task should be located in the upper left part of its window and the least important at the bottom.

❑ **8.2.1.10.2 Task-critical information.** If a window contains task-critical information, that information should be displayed in a way that users can identify easily, for example, separating it from other information by blank space.

8.2.1.11 Visual and audible coding

❑ **8.2.1.11.1 Visual coding of critical information.** A user's attention should be drawn to critical information by highlighting,

color coding, or other means. Neither color coding nor upper-case letters should be used as the sole cue.

■ **8.2.1.11.2 Flash coding.** If flash coding is used, it shall conform to the criteria and guidelines in section 8.5.4.6.

■ **8.2.1.11.3 Audible coding of critical information.** If audible coding is used, it shall conform to the criteria and guidelines in section 8.5.4.3.

8.2.1.12 Dynamic information in windows

□ **8.2.1.12.1 User control.** If a window contains automatically changing information, the application should provide users the ability to: (1) control the rate of update, (2) freeze the display of information, (3) resume updating from the time of freezing, and (4) resume updating from the current time.

□ **8.2.1.12.2 Rate of updating.** If automatically changing information must be read reliably and accurately, the rate of update should not be more than once a second. If users must identify the rate of change or read only gross values, the rate of update should be from two to five times a second (see paragraphs 8.5.7.1.2 and 8.5.7.1.3).

□ **8.2.1.12.3 Dynamic information in frozen, inactive, and iconized windows.** Applications should notify users of critical information that becomes available in frozen, inactive, and iconized windows. If a dynamic window is frozen temporarily, for example, while executing a print command, the application should notify the user immediately of any critical changes and prompt the user to return to automatic updating.

8.2.2 Operations

8.2.2.1 General

□ **8.2.2.1.1 Screen saver.** Systems should provide a screen saver that blanks VDT screens or displays a message or graphic display that changes periodically. Screen savers should be activated when a screen has been idle for three minutes and deactivated when any new activity is detected. The time activation of the screen saver should be user selectable. A new activity includes pressing any key on the keyboard or moving a pointing device.

Exception. Displays containing screens such as a constant monitor screen or one in which users must track an activity over a period of time should not be equipped with a screen saver mode.

8.2.2.2 System log on and log off

□ **8.2.2.2.1 Log on screen.** If a system uses a log on procedure, a log on screen should be displayed automatically as soon as a user completes any required start-up or power-up procedures (same as paragraph 8.3.9.1.2).

■ **8.2.2.2.2 Log on prompts.** If a system log on procedure includes both an identification component (such as a user's name) and an authentication component (such as a user's password), the system shall provide a self-explanatory prompt for each component, with each prompt on a separate line.

■ **8.2.2.2.3 Echoing of user's name, non-echoing of password.** If a log on procedure includes the entry of a user's name and a password, the system shall echo the user's name, but shall not echo the password on the screen.

■ **8.2.2.2.4 Error messages.** If a user makes an error during the log on procedure, the system shall display an error message in the system message area or in a standard pop-up error window. The message shall provide guidance on how to correct the error, but shall not provide information that could assist someone trying to break into the system.

■ **8.2.2.2.5 Completion of log on.** If applicable, upon completion of a log on, the system shall display a main menu or an application window.

■ **8.2.2.2.6 System log off.** A user shall be able to log off a system by selecting the "log off" option from a system-level menu. This menu shall be available at all times the user is logged on.

■ **8.2.2.2.7 Prompting to exit an application or save entries.** If an application is running, and the user initiates a system log off, the system shall prompt the user to save any unsaved entries and exit the application.

■ **8.2.2.2.8 Confirming a log off.** The system shall prompt the user to confirm a log off request.

■ **8.2.2.2.9 Completion of log off.** After completing a system log off, the system shall display the initial system log on screen.

8.2.2.3 Application log on and log off

An application available in a system may require its own log on and log off procedures.

■ **8.2.2.3.1 Log on.** If an application log on is required in addition to the system log on, it shall conform to the same rules as system log on (see section 8.2.2.2).

■ **8.2.2.3.2 Log off.** Logging off an application shall be accomplished with an "exit" function. This function shall be available to users at all times they are logged on to the application.

■ **8.2.2.3.3 Confirming an exit.** The system shall prompt the user to confirm an application exit request.

■ **8.2.2.3.4 Preserving unfinished work.** If the application contains unsaved inputs when the log off request is made, the application shall prompt the user to (1) save the work, (2) confirm the log off, or (3) cancel the request.

■ **8.2.2.3.5 Completion of log off.** Logging off an application shall result in the removal of all screens associated with that application. If one or more other applications are running, the next most current application shall be displayed. Otherwise, the system main menu shall be displayed.

8.2.3 Characters and line length

■ **8.2.3.1 Capitalization.** The United States Government Printing Office *Style Manual* shall be used as a guide for capitalization. If the *Style Manual* does not provide the guidance needed, Merriam Webster's *New International Dictionary* shall be used (same as paragraph 10.2.3.10.1).

■ **8.2.3.2 Capitalization of phrases for emphasis.** In general, capitalization shall not be used to emphasize phrases or sentences (see section 10.3.3.7 for the recommended ways to emphasize text) (same as paragraph 10.2.3.10.2).

> **Discussion.** Continuous text is easiest to read and comprehend when it is presented in mixed case letters. Single words are recognized better when printed in all upper case letters. Thus, if used sparingly and wisely, capitalization can be used to indicate to readers that a word has special significance.

□ **8.2.3.3 Spacing between characters.** Spacing between characters should be at least 10 percent of character height (same as paragraph 7.2.4.7.7).

■ **8.2.3.4 Spacing between words.** Spacing between words shall be at least one character width for equally-spaced characters or the width of capital N for proportionally-spaced characters (same as paragraph 7.2.4.7.8).

■ **8.2.3.5 Spacing between lines.** Spacing between lines shall be at least two stroke widths or 15 percent of character height, whichever is greater. This space is in addition to any space required for accent marks on upper case characters and descenders on lower case letters (same as paragraph 7.2.4.7.9).

> **Note.** The interline spacing recommended for text displayed on terminals is greater than that recommended for printed material (see paragraph 10.3.3.3.1).

■ **8.2.3.6 Spacing between paragraphs.** Paragraphs shall be
separated by a blank line.

□ **8.2.3.7 Minimum character height.** The minimum height of
displayed characters should be 1/200 of the viewing distance.
For example, for a viewing distance of 800 mm (31.5 in),
characters should be at least 4 mm (0.16 in) high (same as
paragraph 8.3.10.4.3).

> **Discussion.** In some cases, a screen will be read by a
> person standing behind a seated user. The character
> height would be based on the maximum viewing distance,
> not the "normal" or "typical" distance.

■ **8.2.3.8 Character width.** The ratio of character height to width
shall be:

a.   1:0.7 to 1:0.9 for equally-spaced characters and lines of
80 or fewer characters,

b.   at least 1:0.5 if it is necessary to have more than 80
characters per line, or

c.   as much as 1:1 for characters such as M and W for
proportionally-spaced characters (same as paragraph
7.2.4.2.5).

□ **8.2.3.9 Stroke width.** Stroke width should be 10 to 12.5 percent
of character height.

□ **8.2.3.10 Minimum dot matrix.** If characters are formed using a
dot matrix, the matrix should be at least 7 dots wide and 9 dots
high.

8.2.4 Color

As emerging information management and control systems
implement graphical user interfaces and high resolution color
graphics displays, color has become a prominent coding method.
Color can reduce clutter in screens and can improve a user's
performance. But, if used inappropriately, color can also induce
the very clutter and performance degradation it attempts to
reduce. For these reasons, color in a display must be used very
carefully. Designers need to keep in mind that:

a.   Both brightness and type of lighting can affect the way
colors are perceived. For example, bright ambient light
desaturates display colors, leading to degraded color
identification and discrimination. In effect, similarly
colored objects can appear different under different
lighting conditions. And conversely, dissimilar colors can
appear similar under different lighting conditions.

b.   The color of a foreground object is affected by the color
of the background.

c.   Visibility and readability are affected by the contrast
between the foreground and the background.

8.2.4.1 Color selection

■ **8.2.4.1.1 General principles.** The following general principles shall be applied to the use of color in screens especially when the color deficiency in the user population is unknown.

a.   Color shall be used sparingly as an information discriminator.

b.   Colors shall be used consistently within a screen and across a set of screens within an application.

c.   The meanings of colors used shall be consistent with user expectations.

d.   Color shall be redundant to another form of coding, such as shape.

e.   To the extent possible, color coding shall be standardized across applications.

□ **8.2.4.1.2 When to use color.** Color should be used to:

a.   augment a user's understanding of the information being presented,

b.   attach specific meaning to a portion of text or a symbol,

c.   direct a user's attention to something,

d.   help a user differentiate rapidly among several types of information,

e.   increase the amount of information conveyed, and

f.   indicate changes in status.

□ **8.2.4.1.3 Constraints on the use of color.** The use of color should avoid or minimize difficulties for users having impaired color vision. Some ways to minimize such difficulty are listed here.

a.   Color should only be added after the effectiveness of a screen has been maximized in an achromatic format.

b.   If similar hues are used, they should be used only with logically related information.

c.   If color is used to emphasize information, the brightest color should be used for the most important information.

d.   The use of color should not reduce screen readability.

■ **8.2.4.1.4 Discrimination of colors.** The colors selected for coding on a screen shall be easily discriminated in all expected operating conditions.

Exhibit 8.2.4.1.4 Discriminable colors and their wavelengths

Color	Wavelength (millimicrons)
Red	700
Orange	600
Yellow	570
Yellow-green	535
Green	500
Blue-green	493
Blue	470

Discussion. Exhibit 8.2.4.1.4 lists some colors (and their wavelengths) that differ enough to permit easy discrimination.

□ **8.2.4.1.5 Colors for infrequently used information.** Shorter wavelength colors (such as blue and green) should be used to display information that is used infrequently.

□ **8.2.4.1.6 Colors for action and status.** Longer wavelength colors (such as red and orange) should be used to suggest action or a demand for a response. Shorter wavelength colors (such as green and blue) should be used to display status or background information.

■ **8.2.4.1.7 Consistent use.** Color shall be used consistently from screen to screen and from application to application.

□ **8.2.4.1.8 One meaning per color.** Each color should represent only one category of displayed data.

■ **8.2.4.1.9 Consistent with conventions.** Color coding shall be consistent with conventional associations with particular colors. For example, red is conventionally associated with danger and power on, and yellow is conventionally associated with caution.

□ **8.2.4.1.10 Number of colors to use.** Color should be introduced into screens conservatively, using relatively few colors to designate critical categories of displayed data, and only if it will facilitate user understanding or performance.

□ **8.2.4.1.11 Maximum number of colors.** The total number of colors used should not exceed four for a single alphanumeric screen and seven for a set of related screens.

□ **8.2.4.1.12 Additional colors.** Additional colors (more than four) should be reserved for special use (for example, in map displays).

Discussion. Only eight or nine highly saturated colors can be easily discriminated.

□ **8.2.4.1.13 Adjacent colors.** Colors displayed next to each other should conform to the following rules:

a.   Highly saturated colors with significantly different wavelengths (such as those toward opposite ends of the spectrum) should not be used next to each other; examples are red and blue, yellow and purple, and magenta and green.

> **Definition. Saturation** is the relative amount of whiteness in a chromatic color.

b.   To convey similarity, similar colors should be used; examples are orange and yellow, and blue and violet.

c.   The foreground color should contrast highly with the background color.

> **Definition. Contrast** is the perception of the difference in the intensity of two areas.

□   **8.2.4.1.14 Color and ambient illumination.** Red should be used only if high ambient illumination is expected. Green and yellow should be used if a broad range of illumination is expected.

□   **8.2.4.1.15 Color key.** If the use of color is extensive or unusual, or if a display may be used infrequently, the display should include a color key that explains the meanings of the colors. If used, a color key should be readily accessible, that is visible without the user having to scroll or expand the display. A color key should include the actual colors being defined.

□   **8.2.4.1.16 Colors at the periphery of large screen displays.** Red and green should not be used at the periphery of large screen displays. Yellow and blue are good colors for use in this area.

■   **8.2.4.1.17 Limiting user color settings.** If a VDT will be shared by different users, individual users shall not be allowed to change those colors in an application that involves color coding or status.

■   **8.2.4.1.18 Easy return to default color scheme.** If users are allowed to change color settings of aspects of an application that do not involve coding, the application shall provide an easy way to restore the default color scheme.

■   **8.2.4.1.19 Portable applications.** If an application is likely to be used on different hardware configurations, it shall be able to accommodate the possible differences in color representations in the different configurations. Status colors shall be assigned during installation, and users shall not be allowed to change them.

■   **8.2.4.1.20 Text-background contrast.** The contrast between text and its background shall be sufficiently high to ensure readability of the text. In general, the color foreground shall differ from its background by a minimum of 100 $_\Delta$E (CIE Yu' v') distances. Luminance contrast ratios for a variety of tasks and conditions shall not be less than those given in exhibit 8.2.4.1.20.

Exhibit 8.2.4.1.20 Luminance contrast ratios for various conditions

Condition	Ratio of foreground to background
Bright ambient illumination	> 7:1
Dark ambient illumination	3:1 to 5:1
To attract attention	> 7:1
To sharpen edges	> 7:1
Continuous reading	3:1 to 5:1
Camouflage images or smooth edges	< 3:1

■ **8.2.4.1.21 Green, yellow, and red.** If green, yellow, and red are used, they shall be used in combination with other cues, such as brightness and saturation, to enhance their discriminability.

□ **8.2.4.1.22 Blue.** Blue should not be used as the foreground color if resolution of fine details is required. Blue is acceptable as a background color.

□ **8.2.4.1.23 Colors for comparison.** If a user must compare data (such as those contained in graphs) based on color, green, yellow, and red should be avoided as comparison colors for application information requiring important or frequent discriminations. If possible, the combinations yellow and blue or red and cyan should be used.

■ **8.2.4.1.24 Small areas.** Users shall not have to discriminate among colors in small areas of the display. If small areas must be coded, they shall be coded achromatically.

■ **8.2.4.1.25 Highlighting.** Highlighting to draw attention to portions of a screen shall be as follows:

a. White highlighting shall be used to draw attention to particular data. When the background is dark, white highlighting shall be used with dark letters. When the background is light, dark highlighting shall be used with white letters.

b. The size and number of areas highlighted shall be minimized.

8.2.4.2 Tonal color coding

This section contains criteria and guidelines about the use of tonal color and shading.

Definition. Tonal coding is coding based on different shades of the same hue or different patterns or textures.

□ **8.2.4.2.1 When to use.** Tonal coding should be used to show relative values of a single variable.

□ **8.2.4.2.2 Ordered coding.** If tonal coding is used to display relative values of a variable, the lightest shade should correspond to the smallest value, and the darkest shade should correspond to the highest value.

8.2.4.3 Color-coded symbols

■ **8.2.4.3.1 Code symbol, not text.** If color is used to indicate status changes, the text itself shall not change color, rather, a box or other shape adjacent to the text shall change color.

■ **8.2.4.3.2 Symbol size.** A symbol that is color coded shall subtend a visual angle of at least 20 min. For example, at a viewing distance of 800 mm (31.5 in), a symbol would be at least 5 mm (0.2 in) high.

■ **8.2.4.3.3 Symbol brightness.** Color-coded symbols shall have a minimum brightness of 3.43 cd/m^2 (1 fL).

■ **8.2.4.3.4 Refresh rate.** Color-coded symbols shall have a refresh rate that provides no perceptible flicker.

8.3 Windowing

Windows provide a convenient and easy to use means of organizing many of the interactive aspects of a system or application and presenting them to a user.

> **Definition.** A **window** is a rectangular area on the screen that provides a visual means for interaction with an application. Applications also use windows to provide information to the user.

This section contains criteria and guidelines for window components, appearance, and states, for window controls and operations, for menus and text in windows, and for a variety of special purpose windows.

> **Caveat.** Much of the material contained in section 8.6 may be very closely tied to a particular scheme or model for implementing windows and handling window management operations. The scheme being alluded to in any one rule may not be the only way of handling windows, nor is it the only recommended, approved, or acceptable way of doing so. To imply otherwise might violate the intent (if not the letter) of paragraph 4.1.10 of this standard. The authors of this guideline have, to the extent possible, removed guidelines that would have eliminated or restricted a particular window management system.
>
> For example, the OSF/Motif ™, Open Look ™, Apple Macintosh ™, and Microsoft Windows™ window management systems all offer similar, but slightly differing models for accomplishing many of the same windowing functions. To prematurely focus upon and exclusively adopt any single one of these management

systems would do a disservice to the users of this proposed guideline.

However, to simply strike out all such implementation-specific referential paragraphs within this section would result in removing a great deal of potentially helpful or useful design guidance information. The editors of these guidelines have chosen to retain these paragraphs for the potential value they might offer as examples of at least one acceptable method of implementing a windows operating environment.

8.3.1 General

■ **8.3.1.1 Hardware limitations on the use of windowing.** Windowing shall be avoided when the hardware has the following limitations:

a.   small screen size, resulting in frequent manipulation of the screen by the user,

b.   slow processing speed, resulting in slow operation by the computer, or

c.   low screen resolution, resulting in less effective visual coding, especially for map graphics, symbols, and icons.

■ **8.3.1.2 User-specified windows.** When the need to view several different types of data simultaneously, the user shall be able to display and select separate windows on a single CRT screen.

■ **8.3.1.3 Number of allowable open windows.** The number of allowable open windows shall not compromise system response time.

> **Discussion.** Each open window requires system resources in terms of memory and processing speed. A limit on the maximum number of windows that can be effectively opened for each system needs to be predetermined.

8.3.2 Window components and appearance

8.3.2.1 General

This section applies to window management systems that have the capability to display primary and secondary windows.

> **Definitions.** A **primary window** is a top or high level window in an application. A **secondary window** is a window that is displayed from within a primary window or another secondary window. Secondary windows are also called "child" windows.

■ **8.3.2.1.1 "Primary" windows.** A "primary" window shall contain: (1) a title bar, (2) a border, and (3) a window menu

control, and (4) a working area. It may also contain a menu bar, controls, objects, and icons.

■ **8.3.2.1.2 "Secondary" windows.** A "secondary" window shall contain: (1) a title bar, and (2) a working area. Secondary windows may contain any of the other window components appropriate to the application.

▢ **8.3.2.1.3 Secondary window constraints.** Secondary windows should be subject to the following constraints:

a. A secondary window should be associated with a particular primary or secondary window.

b. When present, a secondary window should appear within the borders of and on top of (superimposed on) its "parent" window.

c. Closing a secondary window should not affect the parent window.

d. A secondary window should be removed when its parent window is removed.

■ **8.3.2.1.4 Window placement.** Each primary and secondary window shall have a default location defined by the application, at which the window appears when it is first opened. If a window has been moved or resized or both, and is then closed and reopened during an application session, it shall reappear in the size and location it had when it was closed. At the next application, it shall appear in its default location.

8.3.2.2 Title bar

■ **8.3.2.2.1 Description.** A title bar shall appear as a rectangular area at the top of a window, inside the window border with the title of the window in the center; it may contain: (1) a control at the left end that when activated produces a menu of window management options and (2) **Iconize** and **Maximize** controls at the right end.

8.3.2.3 Border

■ **8.3.2.3.1 Description.** Primary and secondary windows shall have a border that encloses all of the window components.

8.3.2.4 Menu bar

(Current versions of Apple Macintosh™ and Open Look™ do not support complete interchangeability between a pointing device and the keyboard for navigation to or within a menu bar.)

■ **8.3.2.4.1 Navigating to the menu bar.** Users shall be able to access the menu bar using both the pointing device and the keyboard. If this is done with the pointing device, the option nearest the pointer when the device button is clicked shall be highlighted. If it is done with the keyboard, the first (leftmost option) shall be highlighted.

Definition. A **location cursor** is an indication of the object in a window that has input focus (see paragraph 8.3.4.3.1). Its shape depends on the type of object; often it is a rectangle that outlines or highlights the object.

■ **8.3.2.4.2 Navigating within a menu bar.** Users shall be able to move the location cursor within a menu bar using both the pointing device and the keyboard. Using the keyboard, users shall be able to move the cursor by pressing the left and right arrow keys, with movement wrapping from the first and last options and may include the system menu of the primary and secondary windows.

■ **8.3.2.4.3 Selecting an option in a menu bar using its mnemonic.** Users shall be able to select an option in the menu bar of a window that has input focus (see paragraph 8.3.4.3.1) by typing the mnemonic (same as paragraph 8.3.7.2.4).

■ **8.3.2.4.4 Leaving the menu bar.** Users shall be able to move the location cursor from the menu bar using both the pointing device or the keyboard. If this is done with the pointing device, the location cursor shall move to the object nearest the pointer when the device button is clicked. If it is done with the keyboard, the location cursor shall return to the object that had input focus before the cursor was moved to the menu bar.

■ **8.3.2.4.5 Displaying a pull-down menu.** Users shall be able to display the pull-down menu for a menu bar option using either the pointing device or the keyboard. When the pull-down menu appears, the location cursor shall be placed on the first option.

□ **8.3.2.4.6 Selecting the default option on a pull-down menu.** Users should be able to select the default option on a pull-down menu by double-clicking on the menu bar option. This should result in the selection of the default option on that menu without the display of the menu.

8.3.3 Window controls

This section contains criteria and guidelines for window controls.

Definitions. A control is any object that allows a user to perform an action. Controls include buttons, menu options, settings, sliders, text fields, and check boxes. A **push button** is a control that appears as a bounded area (for example, a rectangle or oval) on a window. Examples of common push buttons are **OK, Cancel,** and **Help.** Examples of the actions push buttons perform are initiating a command, displaying a pop-up window, and displaying a menu.

8.3.3.1 General

■ **8.3.3.1.1 Consistent and distinctive.** Each type of control in an application window shall be consistent and visually distinct from other types of control. For example, push buttons shall be consistent and distinct from radio buttons (exclusive button sets).

■ **8.3.3.1.2 Distinct from other objects.** Controls shall differ in appearance from other text and graphics in an application window.

8.3.3.2 Text fields

■ **8.3.3.2.1 Applicable criteria and guidelines.** The fields in windows shall conform to the criteria and guidelines for fields in forms given in section 8.4.2.

■ **8.3.3.2.2 When to use.** If a user must be able to type input from the keyboard, a text field shall be provided.

■ **8.3.3.2.3 Scrolling fields.** If a text field will accept more text than can be displayed in the field, a scroll bar shall be provided to enable users to see the entire text. If the anticipated text is expected to exceed a single line, the text field shall be large enough to view multiple lines simultaneously.

8.3.3.3 Scroll bars

■ **8.3.3.3.1 When to use.** If a textual or graphic entity exceeds the space available to display it, a mechanism such as a vertical scroll bar, a horizontal scroll bar, or both shall be provided to enable users to view the entire entity.

■ **8.3.3.3.2 Scroll bar components.** A scroll bar shall contain:

a.   a symbol, such as a box or rectangle, that represents the portion and location of the entity currently displayed,

b.   a vertical or horizontal line or area along which the current display symbol can move, the length of which represents the entire entity, and

c.   two symbols, one above and one below the current display symbol, that allow a user to step through the entity a unit at a time.

□ **8.3.3.3.3 Current display symbol.** The ratio of the length of the current display symbol to the length of the line or area along which it moves should equal the ratio of the portion of the entity displayed to the entire entity.

■ **8.3.3.3.4 Required scroll bar operations.** Users shall be able to operate the scroll bar in at least the following two ways:

a.   Users shall be able to drag the current display symbol continuously along its line or area using a pointing device.

b.   Users shall be able to step through the entire entity in appropriate units, for example, a screen at a time.

□ **8.3.3.3.5 Recommended scroll bar operations.** Users should be able to move directly to a desired position in the window by moving the pointer to a location of the line or area along which

the current display symbol moves and clicking the appropriate device button.

8.3.4 Window states

8.3.4.1 Open, closed, iconized

■ **8.3.4.1.1 Open window.** An open window shall be capable of receiving input from the system. A window that is open and active (see paragraph 8.3.4.2.1) shall be capable of receiving input from a user. An open window shall be completely visible on the screen at the time it is opened and when it is active.

> **Discussion.** More than one window can be open on a screen at the same time. An open window may be partially or totally obscured by another open window, that is, an open window may or may not be visible.

■ **8.3.4.1.2 Closed window.** A closed window shall have no appearance on the screen, neither as a window nor as an icon.

■ **8.3.4.1.3 Closing a primary window.** When a primary window is closed, it and any of its secondary windows shall be removed from the screen. If the window had input focus, the user shall explicitly select another window to have focus; the application shall not arbitrarily assign focus to another window on the screen unless emergency action is required.

□ **8.3.4.1.4 Closing a secondary window.** When a secondary window is closed, it and any of its secondary windows should be removed from the screen. The parent window should not be affected except for the disappearance of the secondary window.

■ **8.3.4.1.5 Iconized windows.** If a user iconizes an open window, the window and any open secondary windows shall be replaced by the window's icon. Any processing occurring in the window may continue.

> **Definition.** To iconize a window is to convert it from a window to an icon (see paragraphs 8.3.5.9 and 8.3.5.10).

■ **8.3.4.1.6 Restoring an iconized window.** It shall be possible to restore an iconized window by using the pointing device, if available (see paragraph 8.3.5.11), and by using the keyboard (see paragraph 8.3.5.12). When the primary window appears, it and all secondary windows associated with it that were open when it was iconized shall be displayed.

8.3.4.2 Active, inactive

■ **8.3.4.2.1 Active windows.** Only one window at a time shall be "active." That window shall have input focus (see paragraph 8.3.4.3.1), and it shall be completely visible, that is, it shall not

be obscured by any other window or icon. An active window shall be distinguishable from inactive windows.

> **Exception.** Complex situations may occur where one window has input focus for keyboard and mouse inputs and another window has input for voice entries.

■ **8.3.4.2.2 Making a window active.** Users shall be able to make any open window active using the **Previous window** operations.

■ **8.3.4.2.3 Making a window inactive.** A window shall become inactive when the user makes any other window active.

> **Discussion.** An inactive window continues to be displayed on the screen, but may be obscured by other windows.

8.3.4.3 Input focus

(Current versions of Apple Macintosh™ do not support the ability to assign input focus with the keyboard.)

■ **8.3.4.3.1 One input focus.** Regardless of the number of windows open in an application, only one window at a time shall be able to receive input from the keyboard.

> **Definition. Input focus,** also called keyboard focus, is the notion that only one window, and usually only one object in a window, at a time is capable of accepting input from the keyboard. Input focus can be explicit (the user must move the pointer into the window and click the appropriate mouse button), or implicit (the user must only move the pointer into the window).

■ **8.3.4.3.2 User assignable input focus.** Users shall be able to assign input focus to any open window of the current application, either with a pointing device or from the keyboard.

■ **8.3.4.3.3 Assigning input focus with a pointing device.** Users shall be able to assign input focus to any window that is wholly or partially visible by moving the pointer onto any visible portion (and clicking the appropriate button, where explicit input focus is necessary). If any portion of the window was obscured by another window, the window with focus shall be made wholly visible.

■ **8.3.4.3.4 Assigning input focus with the keyboard.** Users shall be able to assign input focus to any open window by moving "forward" or "backward" one window at a time through the open windows. For example, users shall be able to press a single key or specific key combinations to move the focus forward or backward one window. Open windows shall be accessible in this way in the order in which they were opened.

■ **8.3.4.3.5 Single object focus.** Only one object in the window having input focus shall be able to receive input from the keyboard. The location cursor or highlighting shall indicate the object having focus. When a window receives input focus, the

location cursor or highlighting shall be placed on either (1) the object that last had focus or, if applicable, (2) the object the pointer was on when focus was achieved.

■ **8.3.4.3.6 Location cursor.** If an object has input focus, that object shall be indicated with a location cursor or highlighting. When a window first appears, the location cursor or highlighting shall be placed on the object users are most likely to select, for example, a text field or a default push button. If a window has lost and then regained input focus, the location cursor or highlighting shall be placed on the object that last had input focus in the window. A user shall be able to move the location cursor or highlighting among objects in the window using either the pointing device or the keyboard.

> **Discussion.** The shape of the location cursor depends upon the nature of the object; sometimes, it is a rectangular box that surrounds the object.

■ **8.3.4.3.7 Assigning input focus to an object** Users shall be able to assign input focus (the location cursor) to an object within a window using either the pointing device or the keyboard.

■ **8.3.4.3.8 Moving input focus to an object with a pointing device.** Users shall be able to move input focus (the location cursor) among objects within a window by moving the pointer onto an object (and clicking the appropriate button, where explicit input focus is necessary).

8.3.4.4 Window mode   Windows may be either "modal" or "modeless."

> **Definitions.** If a **modal** window is on display, a user must interact with that window before he or she can interact with other windows. If a window is **modeless**, a user can interact with other windows.

■ **8.3.4.4.1 Primary window mode.** The primary application window shall be modeless, that is, users shall be able to interact with other windows.

■ **8.3.4.4.2 Secondary window mode.** Secondary windows shall be either modal or modeless. If a window is modal, a user shall not be able to interact with other windows as long as it is displayed. That is, the user must interact with the modal window before being able to interact with any other. If it is modeless, a user shall be able to interact with other windows. The scope of the inability to interact with other windows while a modal window is displayed shall be determined by the application and may extend to: (1) the parent window, (2) all other windows in the application, or (3) all other windows on the screen.

8.3.5 Window operations   For each system or application, the window operations that are performed needs to be identified and their manner of execution made consistent throughout the system. This means that a "standard" way to execute an operation must be available. It is not meant to prohibit developers from providing additional

approaches, for example, providing one method for novices and another for experts. (Current versions of Apple Macintosh™ do not support the ability to move, resize, or iconize windows using a keyboard.)

■ **8.3.5.1 Restoring window to default size.** Where applicable, the application shall provide a **Restore** operation that enables a user to restore an iconized or maximized window to its default size. This option shall be unavailable when the window is its default size.

> **Discussion.** Some icons may not have an associated window. For example, an icon that provides a DOS prompt will not have a window that can be restored.

■ **8.3.5.2 Move.** Where applicable, the application shall provide a **Move** operation that enables a user to move a window on the screen.

> **Discussion.** In some applications, users are not be able to move all windows. For example, some windows are only advisory in nature, such as the amount of processing time remaining. These types of windows cannot be moved, closed, iconized, or resized by the user.

■ **8.3.5.3 Moving a window with a pointing device.** If a window is movable, and a pointing device is available, a user shall be able to move the window by moving the pointer into the window's title bar, pressing the appropriate button on the pointing device, and dragging the window to its new location. The window or an outline of the window shall move on the screen as the user moves the pointing device. Releasing the button shall result in the display of the window in the new location.

■ **8.3.5.4 Moving a window using the keyboard.** Users shall be able to move movable windows using the keyboard by selecting the **Move** option from the window menu (the pointer will change to a move pointer, see exhibit 8.8.3.6.1) and then moving the window or an outline of the window using the arrow keys. Pressing the **Enter** key shall result in the display of the window in the new location.

■ **8.3.5.5 Resize.** Where applicable, the application shall provide a **Resize** operation that enables a user to change the size of a window (see "discussion" in paragraph 8.3.5.2).

■ **8.3.5.6 Resizing a window using a pointing device.** If a pointing device is available, a user shall be able to resize a resizable window by moving the pointer onto the window's border (the pointer will change to a resize pointer, see exhibit 8.8.3.6.1), pressing and holding the appropriate button on the pointing device, and dragging the border to the desired position. As the user moves the pointing device, the window or an outline of the window shall move with it, indicating the changing size of the window. When the user releases the button, the window

shall be displayed in its new size. Moving the pointer onto an edge (top, bottom, or sides) shall permit changing the size in one direction only. Moving the pointer onto a corner shall permit changing the size in two directions at once.

■ **8.3.5.7 Resizing a window using the keyboard.** A user shall be able to change the size of a resizable window using the keyboard by (1) selecting **Resize** from the window menu, (2) selecting the border or corner to be moved using an arrow key or two keys simultaneously, and (3) moving the border or borders using the arrow keys. An outline of the resized window appears after each press of an arrow key. Pressing the **Enter** key shall result in the display of the window in its new size.

■ **8.3.5.8 Iconize.** Where applicable, the application shall provide an **Iconize** operation that changes a window into an icon (see "discussion" in paragraph 8.3.5.2).

■ **8.3.5.9 Iconizing a window using a pointing device.** If the window can be iconized, a user shall be able to change the active window into an icon by moving the pointer onto the **Iconize** control in the title bar, if present, and clicking the appropriate button or by selecting **Iconize** from the window menu.

■ **8.3.5.10 Iconizing a window using the keyboard.** If the window can be iconized, a user shall be able to change the active window into an icon using the keyboard by selecting **Iconize** from the window menu.

■ **8.3.5.11 Restoring an icon using a pointing device.** A user shall be able to restore an iconized window by moving the pointer onto the icon and "double-clicking" (clicking the appropriate button twice at the proper rate of speed). When the window appears, it shall be in the same location and size as when it was iconized. Its status shall be active.

■ **8.3.5.12 Restoring an icon using the keyboard.** A user shall be able to restore an iconized window using the keyboard by (1) moving the location cursor to the desired icon and (2) pressing the **Enter** key. When the window appears, it shall be in the same location and size as when it was iconized. Its status shall be active.

■ **8.3.5.13 Maximize.** If the window can be resized, the application shall provide a **Maximize** operation that enlarges a window to its maximum size.

> **Discussion.** Unless constrained by the application, a maximized window will fill the entire working area of the screen.

■ **8.3.5.14 Close.** If the window can be closed, the application shall provide a **Close** operation that enables a user to close a window, that is, to remove it from the screen. If processing is occurring, or if unsaved data have been generated in the window, users shall

be required to confirm the action before the window is removed
from the screen and processing stops.

□ **8.3.5.15 Next window.** The application should provide a **Next
window** operation that enables a user to assign input focus to the
"next" open window.

> **Discussion.** The concepts of "next" and "previous"
> windows imply an ordering of the open windows. Unless
> required otherwise, the order in which they were opened is
> recommended.

□ **8.3.5.16 Previous window.** The application should provide a
Previous window operation that enables a user to assign input
focus to the "previous" open window.

> **Discussion.** The concepts of "next" and "previous"
> windows imply an ordering of the open windows. Unless
> required otherwise, the order in which they were opened is
> recommended.

□ **8.3.5.17 Moving and copying objects.** Users should be able to
perform the following operations on objects in a window:

a.   move an object to another location in the same window,

b.   move an object to a different window,

c.   copy an object and place the copy at a different location in
the same window,

d.   copy an object and place the copy in a different window.

8.3.6 Window navigation

□ **8.3.6.1 Software navigation aids.** The user should be able to
switch between software modules in a quick, easy manner, using an
interface such as a tree or organization chart. This function
should include the ability to select a menu or submenu directly,
without going through intermediate steps.

□ **8.3.6.2 Open window map.** When using an overlapping window
structure, applications should provide a user-requested iconic or
text map indication of all open windows to allow the user to
easily identify all open (especially hidden) windows.

□ **8.3.6.3 Active designation from open window map.** Users
should be given the capability to designate the active window
through the iconic or text open window map by highlighting the
window representation.

□ **8.3.6.4 Expanded window explanation of open window map.**
If possible, the user should be able to query an open window map
for expanded information (such as the date it was created, its

size, or a description of the subject or application) on the file or application operating in the window.

▫ **8.3.6.5 Window forward function with window map.** When an iconic or text map is provided for determining the numbers and names of open windows in an overlapping system, the user should be able to bring a window forward from the map without having to resize or move other windows.

8.3.7 Menus

8.3.7.1 General

■ **8.3.7.1.1 Applicable criteria and guidelines.** Menus, options, and selection in windows shall conform to the criteria and guidelines in section 8.4.1.

▫ **8.3.7.1.2 Wording of options.** Options should be:

a.   phrased as commands to the computer rather than as questions to the user,

b.   in the vocabulary of the user, not that of the developer,

c.   tersely worded, preferably a single word, and

d.   displayed in mixed case letters, with only the first letter of the first word and acronyms capitalized.

8.3.7.2 Mnemonics and keyboard accelerators

▫ **8.3.7.2.1 Mnemonics.** Each option in a menu should have a mnemonic.

■ **8.3.7.2.2 Single letter mnemonic.** The mnemonic for an option shall be a single letter, different from any other mnemonic in the menu. That letter shall be underlined.

> **Definition.** A **mnemonic** is a single letter that a user can type to select an option in a menu when the menu is displayed.

> **Discussion.** The preferred letter is the first letter, however, if that letter is used as another mnemonic in the menu or associated menus, another letter may be used. It is also preferred that the mnemonic for an option use the same letter in the keyboard accelerator (see paragraph 8.3.7.2.5) if there is one and it includes a letter. For example, "S" might be the mnemonic for a "Save" option, and the simultaneous pressing of **Ctrl** and the letter "S" might be its keyboard accelerator.

- **8.3.7.2.3 Selecting an option using its mnemonic.** If the options in a menu have mnemonics and the menu has input focus, a user shall be able to select an option by simply typing its mnemonic.

- **8.3.7.2.4 Selecting an option in a menu bar using its mnemonic.** Users shall be able to select an option in the menu bar of a window that has input focus (see paragraph 8.3.4.3.1) by typing the mnemonic (same as paragraph 8.3.2.4.3).

- **8.3.7.2.5 Keyboard accelerators.** Applications should provide keyboard accelerators (or "hot keys") for frequently selected menu options. When provided, they should appear right-justified on the same line as the option, separated by enough space to appear visually distinct.

  > **Definition.** A **keyboard accelerator** is a key or simultaneous combination of keys that a user can type to select an option in a menu without having to display the menu. Both mnemonics and accelerators are shortcuts that a user can type from the keyboard.

- **8.3.7.2.6 Selecting an option in a menu using its accelerator.** If a menu has accelerators, a user shall be able to select an option in the menu by typing its accelerator.

- **8.3.7.2.7 Case sensitivity of mnemonics and keyboard accelerators.** Mnemonics and keyboard accelerators shall not be case sensitive, that is, upper and lower case letters shall be equivalent.

- **8.3.7.2.8 Consistency of mnemonics and keyboard accelerators.** Mnemonics and keyboard accelerators shall be consistent throughout an application and related applications.

- **8.3.7.2.9 Displaying mnemonics and accelerators.** Mnemonics and accelerators shall be displayed as part of the menu option. Exhibit 8.3.7.2.9 shows one way of indicating mnemonics (the underscored letters) and accelerators (the key combinations at the right).

Exhibit 8.3.7.2.9 Example of mnemonics and accelerators

<u>U</u>ndo	Alt + Backspace
Cu<u>t</u>	Shift + Del
<u>C</u>opy	Ctrl + Ins
<u>P</u>aste	Shift + Ins
Cl<u>e</u>ar	
<u>D</u>elete	Del

8.3.7.3 Pull-down menus

- **8.3.7.3.1 Title.** The title of a pull-down menu shall be the option on the menu bar with which the pull-down menu is associated. It shall be unique in the menu bar, and, to the extent possible, shall describe or identify the options in the pull-down menu.

  > **Definition.** A **pull-down menu** is a menu associated with an option on a menu bar.

□ **8.3.7.3.2 Presentation of options.** The options in a pull-down menu should be displayed one option per line. Thus the menu should be wide enough to accommodate the longest option and its keyboard accelerator, if present (see also paragraph 8.4.1.5.1).

■ **8.3.7.3.3 Ordering and grouping of options.** The ordering and grouping of options in a pull-down menu shall conform to the criteria and guidelines in section 8.4.1.

■ **8.3.7.3.4 Navigation and selection.** Navigation and selection of options in pull-down menus shall conform to the criteria and guidelines in section 8.4.1.

■ **8.3.7.3.5 Pull-down menu options.** The options in a pull-down menu shall be one of five types (see also paragraphs 8.1.11.1.9 and 8.4.1.1.5):

a.   Commands that are executed as soon as they are selected.

b.   Names of windows or forms that will be displayed. These options shall be identified by a special symbol, for example, an ellipsis (...).

c.   Names of other menus. These options shall be identified by a special symbol, for example, an arrow (→) or triangle (▷) that points to the location where the menu will appear.

d.   Sets of exclusive options. These options shall be identified by special symbol, for example, a filled circle (●) for the selected option and an open circle (○) for the unselected options.

e.   Sets of nonexclusive options. These options shall be identified by special symbols, for example, a marked square (☒) for the selected option(s), if any, and an open square (☐) for the unselected option(s), if any.

□ **8.3.7.3.6 Number of options.** The number of options in a pull-down menu should not be more than ten or less than three (same as paragraph 8.2.11.1.6).

■ **8.3.7.3.7 Distinguishing unavailable options.** If a pull-down menu contains options that are temporarily unavailable, the unavailable options shall be displayed but clearly distinguishable from available options. For example, unavailable options might be displayed at reduced intensity ("grayed out") (same as paragraphs 8.1.11.1.8, 8.1.11.2.7, and 8.4.1.1.6).

8.3.7.4 Scrolling menus   This section contains criteria and guidelines for scrolling menus.

Definition. A **scrolling menu** is a menu, usually containing many options, that does not display all of the options at once, but includes a scroll bar that permits the sequential display of all options. Scrolling menus are also called "list boxes" and "scrolling lists."

■ **8.3.7.4.1 Format.** The displayed options in a scrolling menu shall be arranged vertically, with one option per line, and the scroll bar shall be placed at the right of the displayed options. If the menu has a title, it shall appear above the displayed options, and it shall be easily distinguishable from the options.

□ **8.3.7.4.2 Order of options.** The options in a scrolling menu should be ordered in a way that minimizes user navigation. For example, they might be ordered by expected frequency of use or in chronological or other sequential order. If no other order seems appropriate, they should be ordered alphabetically.

□ **8.3.7.4.3 Number of options displayed.** If a menu contains more than ten options, approximately ten of them should be displayed, and a scroll bar should be provided to permit users to see and select the remaining options. If the options are ordered by expected frequency of use, the highest frequency options should be displayed, and the most frequent option highlighted (same as paragraph 8.4.1.1.3).

■ **8.3.7.4.4 Display of all options in a scrolling menu.** All the options in a scrolling menu shall be available for explicit and complete display through scrolling. It shall be obvious to users that there are more options than are visible (same as paragraph 8.4.1.1.4).

> **Discussion.** The presence of a scroll bar may be sufficient to indicate the existence of additional options.

□ **8.3.7.4.5 Search capability.** If a scrolling menu is large, for example, 50 options or more, the application should provide a search capability that would allow users to type a few characters of the option and search for those characters.

□ **8.3.7.4.6 Editing scrolling menus.** If beneficial for task performance, users should be able to edit scrolling menus, adding, deleting, and changing options.

8.3.8 Text edit windows

■ **8.3.8.1 Text cursor.** The text cursor shall be an I-beam in insert mode and a box over a character in replace mode. The text cursor shall flash at a rate between 2 and 5 Hz. If the text object containing the text cursor loses input focus, the cursor shall stop flashing. If the text object regains input focus, the cursor shall return to normal brightness and resume flashing (same as paragraph 8.4.2.4.1).

■ **8.3.8.2 Text cursor location.** When a window first receives input focus, the text cursor shall be placed in the text area where typing is most likely to occur. If the cursor disappears from view when its window loses focus, the cursor shall reappear at the same location when the window regains focus (same as paragraph 8.4.2.4.2).

■ **8.3.8.3 Moving the text cursor.** Users shall be able to move the text cursor within and among text entry areas using both the pointing device and the keyboard. If a user moves the pointer into a text entry area and clicks the appropriate button, the text cursor shall appear at the pointer location (same as paragraph 8.4.2.4.3).

■ **8.3.8.4 Text cursor display.** The pointer shall change to an I-beam (text cursor) only when the pointer moves into an area in which text entry is possible. Users shall not be able to move the text cursor into areas in which text entry is not possible.

□ **8.3.8.5 Pointer visibility.** The pointer should disappear when a user begins typing and reappear when the user stops typing or when he or she moves the pointing device.

□ **8.3.8.6 Insert mode.** Insert should be the default text entry mode; the **Backspace** key should delete the character to the left of the text cursor; and the Delete key should delete the character to the right of the cursor.

□ **8.3.8.7 Manipulating text.** Users should be able to highlight blocks of text and perform such operations as moving, copying, and deleting on the blocks.

■ **8.3.8.8 Text entry.** Text entry shall be possible only when the text cursor is visible in a location that can accept text entry.

8.3.9 System-level windows

8.3.9.1 System log on and log off

■ **8.3.9.1.1 System access through log on process.** If necessary, each system shall implement a log on procedure that users must complete before they can access any system functions. If the system is unavailable for log on, it shall display a message, if possible, stating the system status and when it will be available.

> **Discussion.** Systems may restrict the applications available to a user based on the users log on identification. Alternatively, systems may require users to log on to individual applications or groups of applications.

□ **8.3.9.1.2 Log on screen or window.** If a system uses a log on procedure, a log on screen or window should be displayed automatically as soon as a user completes any required startup or power-up procedures (same as paragraph 8.2.2.2.1 for screens).

■ **8.3.9.1.3 Log on procedure.** The log on procedure shall conform to paragraph 8.2.2.2.2.

□   **8.3.9.1.4 System startup.** During system startup, the system should display a message stating its unavailability, change the pointer shape to a watch or hourglass, and disable the keyboard and pointing device. When startup is complete, the message should be removed, the pointer returned to its normal shape, and the keyboard and pointing device enabled. If appropriate, the system should provide messages containing such information as average system response time or known periods of unavailability.

■   **8.3.9.1.5 User-initiated log off.** Users shall be able to initiate a log off by selecting **Log off** from the system menu (see section 8.3.9.3). If a user has generated any data without saving them, the system shall prompt him or her to save the data, if desired, and confirm the log off. If applicable, the system shall notify the user of any applications that are still running and require the user to log off the application(s) before confirming a system log off. When log off is complete, the initial log on window shall be displayed.

□   **8.3.9.1.6 Automatic log off.** If a system includes an automatic log off due to user inactivity, a standard elapsed time (for example, 15 minutes) should be designated for automatic log off. This time interval should be modifiable by the user. During periods of inactivity, the system should display a message stating the action necessary to avoid automatic log off (for example, a keystroke or movement of the pointing device). In addition, an auditory signal should be provided at intervals during the period of inactivity. If automatic log off occurs, any unsaved data should be saved, and a message should be displayed indicating that automatic log off has occurred and providing the name of the file in which data have been saved, if applicable.

8.3.9.2 The system window

■   **8.3.9.2.1 Appearance.** The system window shall appear when system startup is complete. It shall occupy the entire screen.

■   **8.3.9.2.2 System window components.** The system window shall contain:

a.   a system title bar that extends across the top of the screen,

b.   a system menu bar located just below the system title bar, and that also extends across the screen, and

c.   an area available for the display of application windows that occupies the remainder of the screen.

The system title bar shall contain a centered title that identifies the system. It may also include optional components such as status indicators and a date and time display. The system menu bar shall list the titles of menus that are available at the system level. These menus shall provide access to the application level programs available to the user. The application area of the

system window may contain icons that represent application windows or action icons common to all applications.

■ **8.3.9.2.3 System window behavior.** Users shall not be able to move or resize the system window, nor shall they be able to obscure the system title bar or system menu bar. Appropriate system-level menu options shall always be available.

8.3.9.3 The system menu

■ **8.3.9.3.1 Integration of applications into system-level menus.** The system designers shall decide whether the functionality provided by an application will be available as an entire menu, as a group of options within a menu, or as a single menu option, and whether the application will be available from the system-level menu or from within another application.

□ **8.3.9.3.2 System window menu bar.** The set of options that appears in the system menu bar should describe the overall functionality of the system. The menu bar should contain no more than ten options plus **Help**. The options should begin at the left margin and extend to the right with enough space between them so that they can be read easily and to accommodate the longest options in the pull-down menus. Each option (and each option in the pull-down menus) should contain a mnemonic to permit selection from the keyboard.

□ **8.3.9.3.3 Consistency across systems.** To the extent possible, menu bar options and their order should be the same across systems. If the same application appears in different systems, it should have the same name in each system and should be available in the same system-level menu.

□ **8.3.9.3.4 Access to Help.** When users are working in an application, they should be able to select **Help** from the system menu bar at any time.

□ **8.3.9.3.5 Navigation aid.** Each system should include a navigation aid that provides an overview of the system and allows users to navigate quickly to a particular part of the system. For example, the system might provide a graphical representation of the system that would allow a user to select one part and have the appropriate window displayed on the screen. This navigation aid should be accessible through **Help**.

8.3.9.4 System support

□ **8.3.9.4.1 System menu.** Each system should provide a system menu that includes options to end a session, print screens, review system status, define user preferences, manage alerts, change a password, access peripherals, and perform file management. These options should be available through a **System** option in the system menu bar.

□ **8.3.9.4.2 Utilities menu.** Each system should include the resources required to support the functionality provided by the system, including such resources as word processing, spreadsheets, and electronic mail. These resources should be available through a **Utilities** option in the system menu bar.

□ **8.3.9.4.3 Additional support functions.** Each system should provide:

    a.   a screen saver,

    b.   the ability to suspend a session without completely logging off (the system would continue all active processes, but not allow interaction until a user logs on again), and

    c.   easy identification of and navigation among all open windows.

□ **8.3.9.4.4 User-specified settings.** System designers should decide which interface parameters users will be allowed to set, provide a default setting for each, and provide users access to these settings. The designers should decide which of these settings will remain in effect for the current session only and which will be in effect whenever that user logs on. Users should be able to review the parameters and reset them at any time during a session. At the end of a session, any parameters having settings that apply only to the current session should be reset to their default values.

8.3.10 Application-level windows

Current versions of Apple Macintosh™ do not support applications windows per se. In the Macintosh windowing system, the system-level window changes its menu bar according to the application that is running. Therefore, there is no distinction between system-level windows and application-level windows in the Macintosh environment.

8.3.10.1 Window organization

■ **8.3.10.1.1 Components of application windows.** All application windows shall have a border or frame, a title bar, a window menu control, and a working area. If appropriate, they may also contain a window menu bar, a command entry area, and a message area. If present, these components shall be located as follows:

    a.   The title bar shall extend across the top of the window.

    b.   The window menu control shall be at the left end of the title bar.

    c.   The menu bar shall extend across the window just below the title bar.

d.   The working area shall occupy all the space inside the border that is not occupied by another component.

e.   The command entry area shall extend across the bottom of the window just above the message area, if there is one.

f.   The message area shall extend across the bottom of the window.

■ **8.3.10.1.2 Window title.** The window title shall appear centered in the window's title bar. It shall be in mixed-case letters, and it shall be unique in the application. The title shall be as informative as possible, describing the purpose of the window, and, if appropriate, including the name of the application. If a window is displayed as a result of the selection of an option in a menu, the title of the window shall be the same as the wording of the option.

□ **8.3.10.1.3 Window menu bars.** Window menu bars should contain no more than ten options plus **Help**. The options should begin at the left margin and extend to the right, with **Help** located consistently. Options in window menu bars should not duplicate options in the system menu bar.

□ **8.3.10.1.4 Names of menu bar options.** Each menu that appears as an option in a menu bar should have a title that is unique in the application. If the same menu occurs in different windows, it should have the same title in each. Each title should have a mnemonic.

□ **8.3.10.1.5 Push buttons.** The top, bottom, or sides of the working area should be reserved for push buttons that provide actions that can be taken in the window. The push buttons should be displayed in a horizontal or vertical row centered with the window. Button order should be consistent throughout an application. Buttons should be ordered from left to right (or top to bottom for vertical rows) according to one or more of the following principles:

a.   by frequency of use, with the most frequent at the left or top,

b.   by sequence of use, with the first to be used at the left or top, or

c.   with positive actions at the left or top and negative or canceling actions at the right or bottom.

□ **8.3.10.1.6 Help button.** If **Help** does not appear in a window menu bar, the window should have a **Help** button. It should be located at the bottom right corner of the working area of the window.

□ **8.3.10.1.7 Action icons.** If an application window includes action icons, they should be arranged along the left margin of the

window. The number of action icons in a window should not exceed 20.

■ **8.3.10.1.8 Action icons bound to window.** If an application window includes action icons, a user shall not be able to move the icons outside the window.

□ **8.3.10.1.9 Message area.** Noncritical application messages to users should be presented in a message area at the bottom of the window. The left side of the area should be used for routine messages, simple help, and status messages. The right side of the area should be used to present information about the window, such as the name of an object or the page number. Primary windows should have message areas.

> **Discussion.** The message area may be a dedicated area or it may be an area that is used temporarily when a message is presented, but is available for other uses otherwise.

■ **8.3.10.1.10 Consistency in window organization.** The windows in an application and related applications shall have a consistent organizational scheme for the key elements of the windows. Individual windows shall contain only those elements appropriate to the particular task, but the elements shall be consistent from window to window throughout the application.

□ **8.3.10.1.11 Control windows (dialog boxes).** Sets of controls that perform similar or related functions should be grouped and presented together in a control window (also called a dialog box). A control window should have a border and a title that clearly indicates the function of the set of controls. The individual controls should be arranged in an orderly and logical manner. If scroll bars are needed, vertical scroll bars should be located at the right, and horizontal scroll bars should be located at the bottom of the area to be scrolled. If a control is temporarily unavailable, it should be displayed at reduced intensity. Exhibit 8.3.10.1.11 is an example of a control window.

8.3.10.2 Message windows

This section contains criteria and guidelines for several special purpose message windows (also called message boxes). These include request windows, error message windows, information message windows, confirmation message windows, warning message windows, and "working" message windows.

> **Definition.** A **message window** is a secondary window that provides users (1) noncritical information, (2) progress information about lengthy processes, (3) alerts to unusual events, and (4) warnings of potential dangers. Message windows may be modal or modeless.

□ **8.3.10.2.1 Allowed operations.** Users should be able to **Move** a message window.

□ **8.3.10.2.2 Disallowed operations.** Users should not be able to **Iconize** or **Resize** message windows.

Exhibit 8.3.10.1.11 Example of a control window

□ **8.3.10.2.3 Message windows -- contents.** Message windows should contain (1) a title, (2) a symbol that indicates the type of message, (3) the message itself, and (4) one or more push buttons.

> **Discussion.** Some examples of possible symbols for different types of messages are: ∅ for error messages, **i** for information messages, **?** for request and confirmation messages, **!** for warning messages, and a watch, clock, or hourglass for "working" messages.

□ **8.3.10.2.4 Message wording.** The messages in message windows should use language that is meaningful to users and should require no further documentation or translation. Messages should focus on what needs to be done, not on what was done wrong.

□ **8.3.10.2.5 Message windows -- size and location.** Message windows should be just large enough to display the information required. They should be distinctive in appearance and appear in a standard location on the screen.

□ **8.3.10.2.6 Request message windows.** A request message window should be used when it is necessary to request information from a user before processing can proceed. A request message window should contain: (1) a title, (2) a question symbol (?), (3) a message indicating the information required, (4) a text field, if appropriate, and (5) all of the following push buttons that apply in the order in which they are listed: **OK, Apply, Reset, Cancel,** and **Help.**

□ **8.3.10.2.7 Error message windows.** An error message window should contain: (1) an error symbol (∅), (2) a message, and (3) the following push buttons in the order listed: **OK, Cancel,** and **Help.**

□ **8.3.10.2.8 Information message windows.** An information message window should be used to convey noncritical information that requires acknowledgement by users. (The message area described in paragraph 8.3.10.1.9 should be used for messages that do not require acknowledgement.) An information message window should contain: (1) an information symbol (i), (2) a message, and (3) the following push buttons in the order listed: **OK** and **Help.**

■ **8.3.10.2.9 Information message window behavior.** Information message windows shall not appear to the user to interrupt processing by the application. That is, if the application interrupts processing, it shall be transparent to the user.

□ **8.3.10.2.10 Confirmation message windows.** Confirmation message windows should be used to request clarification of a previous user action. The application should suspend processing until the user responds to the window. The window should contain: (1) a question symbol (?), (2) a message, and (3) one of the following sets of push buttons in the order listed: {**Yes, No,** and **Help**} or {**Yes, No,** Cancel, and **Help**}.

□ **8.3.10.2.11 Warning message windows.** Critical messages warning users of destructive consequences of actions should be displayed in warning message windows, and processing should be suspended until a user responds. Warning message windows should contain: (1) a warning symbol (!), (2) a message, and (3) one of the following sets of push buttons, in the order listed: {**Yes, No,** and **Help**} or {**OK, Cancel,** and **Help**}.

□ **8.3.10.2.12 Accompanying audible warning signals.** Warning messages should be accompanied by an audible signal (see also paragraph 8.5.4.3).

■ **8.3.10.2.13 "Working" message windows.** If the processing time resulting from a user action will exceed 2 seconds, the system shall display a "working" message window. The display of the window shall not interrupt processing. The message window shall remain on display until processing is completed or until the user iconizes the window or cancels the process. The window shall be removed automatically when processing is completed. Working message windows shall contain: (1) a working symbol (for example, a clock, watch, or hourglass), (2) a message, and (3) one of the following sets of push buttons, in the order listed: {**OK** and **Help**}, {**OK, Cancel,** and **Help**}, {**OK, Stop,** and **Help**}, or {**OK, Pause, Resume, Stop,** and **Help**}.

□ **8.3.10.2.14 Progressive working messages.** If processing time will be lengthy, the window should be updated to indicate the status of processing (for example by displaying messages like

"5% completed") or should include a scale showing the proportion of processing completed.

8.3.10.3 Use of color in windows

■ **8.3.10.3.1 Applicable criteria and guidelines.** The use of color in windows shall conform to the criteria and guidelines in section 8.3.4.

■ **8.3.10.3.2 Consistent use of color.** Color shall be used consistently throughout an application and related applications.

□ **8.3.10.3.3 Limited use of color.** Color should be used in the working area of a window primarily as redundant coding and as a means of highlighting critical elements.

■ **8.3.10.3.4 Coding and status colors.** Users and applications shall not be able to change colors for coding and status of facilities, services, or equipment such as alarms or alerts.

□ **8.3.10.3.5 User preferences.** If appropriate to the functionality of an application, users should have the option of selecting from a variety of color sets as a user preference setting for aspects of an application that do not involve coding or status.

8.3.10.4 Text in windows

□ **8.3.10.4.1 Contrast.** In general, text should be displayed as black characters on a white or light background (same as 8.4.2.1.3 and 8.5.2.1.2).

□ **8.3.10.4.2 Dot matrices.** If characters are formed using dot matrices, the matrix should be at least seven dots wide and nine dots high.

□ **8.3.10.4.3 Minimum character height.** The minimum height of displayed characters should be 1/200 of the viewing distance. For example, for a viewing distance of 800 mm (31.5 in), characters should be at least 4 mm (0.16 in) high (same as paragraph 8.2.3.7).

> **Discussion.** In some cases, a screen will be read by a person standing behind a seated user. The character height would be based on the maximum viewing distance, not the "normal" or "typical" distance.

□ **8.3.10.4.4 User-selectable font size.** If an application cannot satisfy the range of viewing requirements with a single text font, the application should provide text font size as a user-selectable option.

□ **8.3.10.4.5 Arabic vs. Roman numerals.** If information elements in a window will be numbered, Arabic numerals should be used, not Roman numerals.

❑ **8.3.10.4.6 Capitalization.** Text should be presented in a combination of upper- and lower-case letters, following standard capitalization rules (see, for example, the U. S. Government Printing Office *Style Manual*).

❑ **8.3.10.4.7 Acronyms.** Acronyms should be used only if they are significantly shorter than the term they represent and if they will be commonly understood by users. Acronyms should be displayed in all upper-case letters.

❑ **8.3.10.4.8 Abbreviations.** Abbreviations should be used only if they are significantly shorter than the word they represent and if they will be commonly understood by users. They should be as short as possible, consistent with uniqueness in the application.

■ **8.3.10.4.9 Formation of acronyms and abbreviations.** New acronyms and abbreviations shall be formed according to the rules in the U.S. Government Printing Office *Style Manual*.

■ **8.3.10.4.10 Dictionary of acronyms and abbreviations.** If acronyms or abbreviations are used, an on-line dictionary or help screen shall be provided.

❑ **8.3.10.4.11 Consistent structure for noneditable text.** Each type of noneditable text, for example, titles, labels, and instructions, displayed in windows should have a consistent grammatical structure. For example, all instructions might be complete, imperative sentences.

■ **8.3.10.4.12 Vocabulary.** The words used in all noneditable text shall be task-oriented and familiar to users.

❑ **8.3.10.4.13 Sentence structure.** In continuous text, sentences should be simple, affirmative, and active, as opposed to complex or compound, negative, and passive.

> **Exception.** Negative sentences are appropriate for stating prohibitions, for example, "Do not exit the application before saving your entries."

❑ **8.3.10.4.14 Punctuation.** Normal punctuation rules should be followed. Contractions and hyphenation should be avoided.

■ **8.3.10.4.15 Sequences.** Sequences of events or steps shall be presented in the proper order.

■ **8.3.10.4.16 Referents.** The referents for pronouns such as "it" and "they" shall be easily identifiable.

❑ **8.3.10.4.17 Case conversion.** If an application requires that all text be in one case, for example upper-case, the application should accept typed upper- and lower-case letters as equivalent and automatically convert the "improper" case to the "proper" one.

□ **8.3.10.4.18 Wild card search characters.** If an application provides a character string search capability, it should include the following "wild card" characters:

a.   @ should represent any single upper- or lower-case alphabetic character. For example, abc@d would retrieve abcad, abcEd, and abczd; it would not retrieve abc7d or abcd.

b.   # should represent any single numeric character. For example, 123#4 would retrieve 12334, 12394, and 12304; it would not retrieve 123554 or 123A4.

c.   ? should represent any single alphanumeric character (that is, any upper- or lower case alphabetic character, any number, or any punctuation mark). For example, abc?d would retrieve abcAd, abc5d, and abc,d; it would not retrieve abcxxd.

d.   * should represent zero or more alphanumeric characters. For example, abc*d would retrieve abcd, abcad, and abcjf75/kld.

8.3.11 Window management considerations

□ **8.3.11.1 Initial window contents and organization.** The initial contents and organization of a window should permit a user to accomplish the window's purpose easily and efficiently.

□ **8.3.11.2 Initial size.** If possible, the initial size of a window should permit the display of all its contents. The minimum size should permit the display of the title and menu bar, if any.

□ **8.3.11.3 Initial placement.** The initial placement of a window should be based on: (1) the importance of the information (critical information should be placed in the center of the user's field of view), (2) information already displayed that should not be obscured, (3) the distance from the current pointer location (pointer movement should be minimized), and, (4) if applicable, information already displayed that is relevant to the window.

□ **8.3.11.4 Resizing.** When a user resizes a window, only the border(s) affected should move, not the objects within the borders. If a window becomes too small to display its objects, vertical or horizontal scroll bars or both should be added.

> **Discussion.** If appropriate, the size to which a window can be reduced may be restricted so that its objects cannot be obscured.

□ **8.3.11.5 Mode of windows.** Developers should specify the mode (modal or modeless) of each secondary window (see section 8.3.4.4).

8.3.12 Task-specific windows

8.3.12.1 Help windows   Specific guidelines pertaining to on-line help are available in section 8.6.1.

☐ **8.3.12.1.1 Help availability.** Both system-level and application-level help should be available to users. Help should be provided in the following ways:

a.   As a menu title in the system menu bar. This level of help should describe system capabilities and provide information on how to use help. It may include an on-line tutorial for users and a system navigation aid (see paragraph 8.3.9.3.5).

b.   As a menu title in an application menu bar. This level of help should include general information on application functionality. It may include an on-line, cross- referenced index so that users can obtain information about particular windows, actions, and commands. If the application uses action icons, it may provide help through an action icon.

c.   As a push button or check box in a window. This level of help should provide information about the actions that can be taken in the window.

d.   As a message in the message area of a window. This level of help should explain how to complete the initiation of an action.

e.   As a function available from the keyboard. This level of help should provide information about the object in a window that has input focus. The information may be displayed in a message window or in the message area of the window in which the object appears.

☐ **8.3.12.1.2 Help window elements.** A help window should include: (1) a title that identifies the contents, (2) a working area, scrollable, if necessary, that displays the help information, and (3) an **OK** push button to remove the window.

> **Discussion.** It is desirable that users be able to print part or all of the contents of a help window.

☐ **8.3.12.1.3 Size and placement.** Help windows should be wide enough to display complete lines of text and long enough to display all the lines, if practical. The window should be placed so that it does not obscure the object it describes. Users should be able to **Move** and **Resize** help windows.

☐ **8.3.12.1.4 Help information.** A help window should describe the object or explain the steps required to initiate the action about which help was requested.

□ **8.3.12.1.5 Wording of help information.** The appearance and wording of help information should conform to the criteria and guidelines in section 8.6.9.4. In addition, words or terms likely to be of interest to users should be highlighted, and steps should be numbered and presented as separate paragraphs.

□ **8.3.12.1.6 Removal of help windows.** Help windows should be removed from the screen when: (1) a user activates the **OK** push button, or (2) the object or window about which help was requested is removed, iconized, or closed.

> **Discussion.** It is desirable that users be able to keep a help window displayed while continuing to work with the application.

8.3.12.2 Data entry windows

□ **8.3.12.2.1 Data entry window elements.** A data entry window should contain: (1) a title that describes the purpose or contents of the window, (2) a set of labeled fields, (3) vertical or horizontal scroll bars or both, if the contents do not fit in the window's working area, and (4) controls appropriate to the task.

> **Definition. A data entry window** is a window that contains a set of labeled fields for entering, changing, and deleting data. It may also contain labeled data display fields, which a user cannot change.

□ **8.3.12.2.2 Data window organization.** The organization of a data entry window should be consistent with the task it represents. For example, data fields should be arranged by sequence of use, frequency of use, or importance, with related fields grouped together and separated from unrelated fields. If users will enter data from hard copy forms, the data entry field organization should correspond to that of the hard copy.

□ **8.3.12.2.3 Multipage data entry windows.** If the contents of a set of data entry fields do not fit the window's working area,

a. the window should provide users the ability to page, scroll, or both through the entire set, and

b. if the fields are arranged in rows, columns, or both, the labels of the rows or columns should remain in place when the rows or columns scroll or page.

□ **8.3.12.2.4 Push buttons in data entry windows.** If a data entry window contains push buttons, the buttons should be placed in a row at the bottom of the working area, visually separated from the data fields.

□ **8.3.12.2.5 Controls for data entry windows.** A data entry window should contain the controls appropriate to the task.

Examples. If the contents require more than one page, the window would contain controls for paging. It might also be appropriate to include controls for clearing entries and restarting data entry.

☐ **8.3.12.2.6 Help on data entry fields.** Users should be able to obtain help about data entry fields.

■ **8.3.12.2.7 Data entry fields.** Fields in data entry windows shall conform to the criteria and guidelines in section 8.4.2.2.

■ **8.3.12.2.8 Labeling data fields.** The labels for data entry fields shall conform to the criteria and guidelines in section 8.4.3.3.

■ **8.3.12.2.9 Navigation in data entry windows.** Navigation in data entry windows shall conform to the criteria and guidelines in section 8.4.3.5.

■ **8.3.12.2.10 Data entry and editing in data entry windows.** The entering and editing of data in data entry windows shall conform to the criteria and guidelines in section 8.4.3.7.

■ **8.3.12.2.11 Saving entered data.** When a user has finished making entries in a data entry window, he or she shall be able to save the entries by taking an explicit action such as selecting a **Save** menu option or activating an **Apply** or **OK** push button. Users shall be able to save their entries at any time during data entry.

8.3.12.3 Text windows

☐ **8.3.12.3.1 Width of a text window.** A window intended for the display of textual information should be wide enough to display an entire line of anticipated text without horizontal scrolling. The text should automatically wrap to the next line based on the width of the window.

■ **8.3.12.3.2 Vertical scroll bars.** If the entire text document does not fit in the current window, the window shall have a vertical scroll bar or a similar mechanism (positioned either on the right or left side of a window) so that users can view the entire document. The current position in the document (for example, the current page or line number) shall be displayed in a consistent location, such as in the window's message area.

☐ **8.3.12.3.3 Document operations.** As appropriate, users should be able to **Save, Retrieve, Edit, Delete, Print** (all or specified portions), and **Rename** documents.

☐ **8.3.12.3.4 Text manipulation.** If appropriate, users should be able to specify the format of a document (for example, set margins and tab stops) and to select the font type, size, and style for text. Automatic line breaks and page breaks should be provided. Users should be able to assign page numbers as well as have them supplied automatically. If the displayed text is not

formatted as it will be when printed, users should have the ability to view the text in its final format.

□   **8.3.12.3.5 Search and replace capabilities.** Users should have both search and search-and-replace capabilities in text windows.

8.3.12.4 Map windows

□   **8.3.12.4.1 Map window elements.** A map window should include: (1) a title, (2) identifying information such as coordinates, area, and scale, (3) the map itself, (4) a continuous coordinate indicator that states the pointer location, and (5) appropriate controls.

□   **8.3.12.4.2 Map orientation.** All maps should be displayed in the same orientation, usually with north at the top.

□   **8.3.12.4.3 Consistent label position.** Labels should be positioned consistently with respect to the feature they identify, for example, to the left of or below the feature, but without obscuring important information.

□   **8.3.12.4.4 Label legibility.** Labels should remain legible at all map resolutions.

□   **8.3.12.4.5 Color coding key.** If a color overlay is available for a map, a color coding key that explains each color should be displayed whenever the overlay is displayed. Users should be able to display the key without displaying the overlay.

■   **8.3.12.4.6 Symbols and color codes.** Symbols and color codes shall conform to the criteria and guidelines in sections 8.5.4.5 and 8.5.4.8.

□   **8.3.12.4.7 User control of map appearance.** Users should be able to customize a map window to conform to the task being performed as follows:

a.   pan and zoom (see section 8.5.8.3) ,

b.   return to initial appearance,

c.   define a "home" position and return to this position easily,

d.   move the window (see paragraphs 8.3.5.3 and 8.3.5.4),

e.   define the map appearance (for example, assign colors to areas) (see section 8.5.4.5 for color coding),

f.   select the objects that appear on the map, and

g.   change the appearance of critical information.

□ **8.3.12.4.8 User editing of labels and overlays.** If authorized, users should be able to add, edit, reposition, and delete labels and overlays on a map.

□ **8.3.12.4.9 "Reading" a map.** Users should be able to determine the distance and bearing between any two points on a map.

□ **8.3.12.4.10 Crowded, cluttered maps.** If symbols on a map are densely packed or overlapped, users should have a way to select the desired symbol easily and accurately (for example, by selecting it from a pop-up menu). Users should be able to distinguish among symbols that represent coincident points and to obtain information that will allow them to resolve ambiguities among symbols.

□ **8.3.12.4.11 User control of automatic updating.** Users should be able to select the categories of information that will be updated automatically and to specify the frequency and rate at which the information will be updated. If appropriate, users should also be able to temporarily stop and then resume updating.

8.3.12.5 Windows for sending and receiving electronic messages

■ **8.3.12.5.1 Message handling windows.** Windows intended for sending and receiving electronic messages shall conform to the general criteria and guidelines for data entry windows given in section 8.6.12.2.

□ **8.3.12.5.2 Message window fields and headers.** Message handling windows should include a basic set of labeled fields, for example, an addressee field, a "copy to" field, a subject field, and a message field. If appropriate, a variety of preformatted forms corresponding to standard message formats should be provided.

□ **8.3.12.5.3 Field support.** If possible, the application should provide information to help a user make a proper entry in a field, for example, the addressee field might be supported by a pop-up menu of potential addressees.

□ **8.3.12.5.4 Distribution lists.** Users should be able to create, store, retrieve, edit, and use distribution lists of commonly used addressees or groups of addressees.

□ **8.3.12.5.5 Message transmission.** Users should be able to transmit electronic messages easily, for example, by activating a **Send** or **Transmit** push button.

□ **8.3.12.5.6 Delayed or unsuccessful transmission.** If an electronic message cannot be sent immediately, the system should automatically queue the message; users should not have to monitor transmission or make repeated attempts to send a message. Users should be notified if a message cannot be

transmitted. Users should be able to cancel or abort any message that has not yet been transmitted.

□   **8.3.12.5.7 Message status information.** Users should be able to specify what sort of feedback they want about message transmission, and a log of this information should be maintained automatically.

□   **8.3.12.5.8 Notification of message arrival.** Users should be notified when a high priority electronic message arrives. For example, at log on, users might be given a list of messages that have arrived since they last read a message, and during a session, an alert of some sort might be displayed in the system window. Notification should not interfere with ongoing system use. If messages differ in priority, the notification should reflect that priority.

□   **8.3.12.5.9 Queuing and logging incoming messages.** Incoming electronic messages should be automatically queued by time of receipt and message priority, if any, and a log of this information should be maintained.

□   **8.3.12.5.10 Incoming message operations.** Users should be able to display a summary of new electronic messages addressed to them and any old messages they have not deleted. They should be able to **Display, Save,** and **Delete** individual messages. When a message is displayed, it should appear in a text window, with all the capabilities of these windows, such as scrolling and printing.

8.4 Data entry

Data entry refers to user actions involving the input of data into a computer system, and the system's response to the user actions. The data entry methods covered in this section are: (1) selection from menus, (2) form filling, (3) direct manipulation, and (4) the keyboard entry of text. Additional topics covered in this section include the entry of tabular and graphic data, and the validation of entered data.

8.4.1 Menus

Menus are often useful in data entry, for example, to list files that may be retrieved, or to list the acceptable entries for a field in a form. Menus of this sort are often too long to display in their entirety. In that case, a portion of the menu is displayed and a scrolling capability is provided.

8.4.1.1 General

■   **8.4.1.1.1 Consistent style.** Menus throughout an application shall conform to a single style of interface, for example, OSF/Motif™, Open Look™, Microsoft Windows™, or Macintosh™ (same as paragraph 8.1.11.1.2).

■   **8.4.1.1.2 Consistent wording and ordering.** Menus and options that appear in different displays and contexts shall be consistent in wording and order (same as paragraph 8.1.11.1.3).

□ **8.4.1.1.3 Scrollable menus.** If a menu contains more than ten options, approximately ten of the options should be displayed, and a scroll bar or similar mechanism should be provided to permit a user to see and select the remaining options. If the options are ordered by expected frequency of use, the highest frequency options should be displayed, and the most frequent option highlighted (same as paragraph 8.3.7.4.3).

■ **8.4.1.1.4 Display of all options in a scrolling menu.** All the options in a scrolling menu shall be available for explicit and complete display through scrolling. It shall be obvious to users that there are more options than are visible (same as paragraph 8.3.7.4.4).

> **Discussion.** The presence of a scroll bar may be sufficient to indicate the existence of additional options.

■ **8.4.1.1.5 Distinguishing types of options.** If a menu contains options of different types, for example, options that lead to other menus and options that are values that can be entered in fields, the types shall be distinguishable. For example, options that lead to other menus might be followed by a triangle that points to where the subsequent menu will appear (▷ or ▽). A menu option that requires additional information from the user might be followed by an ellipsis (...) (same as 8.1.11.1.9; see also 8.1.11.1.8).

■ **8.4.1.1.6 Distinguishing unavailable options.** If a menu contains options that are temporarily unavailable, the unavailable options shall be displayed but clearly distinguishable from available options. For example, unavailable options might be displayed at reduced intensity ("grayed out") (same as paragraphs 8.1.11.1.8, 8.1.11.2.7, and 8.3.7.3.7).

■ **8.4.1.1.7 Instructions.** Instructions pertaining to menus shall appear in a help window and in a consistent location on the screen (same as paragraph 8.1.11.1.10).

■ **8.4.1.1.8 Menus distinct from other displayed information.** Menus that appear in displays that also contain other objects or information shall be distinct from the other objects or information (same as paragraph 8.1.11.1.13).

8.4.1.2 Hierarchical menus

□ **8.4.1.2.1 When to use.** Hierarchical menus should be used if the number of options is more than ten and the options can be organized into a meaningful hierarchy.

> **Note.** A hierarchical structure may be more cumbersome and keystroke intensive than a longer, single-level structure. Thus, if a long list of options is obviously and logically organized, it will be easier to use than a hierarchical structure. For instance, consider a list of

type sizes numerically ordered or a long list font
alternatives logically organized.

■ **8.4.1.2.2 Applicable rules.** If hierarchical menus are used for
data entry, they shall conform to the rules in section 8.1.11.3.

**8.4.1.3 Pull-down
menus**

Pull-down menus have limited applicability in data entry, but
they may be useful for such activities as retrieving files.

> **Definition.** A **pull-down menu** is a menu that appears
> when a menu bar option is selected.

□ **8.4.1.3.1 When to use.** Pull-down menus should be used rather
than pop-up menus if the position of the cursor on the screen is
not important for information or option retrieval (same as
8.1.11.5.1).

> **Discussion.** The advantage of pull-down menus over
> pop-up menus is that pull-down menus always have a
> visual cue in the form of a menu bar.

■ **8.4.1.3.2 Consistent location.** Pull-down menus shall appear
immediately below or adjacent the option whose selection leads
to their appearance (same as paragraph 8.1.11.5.2).

8.4.1.4 Pop-up menus

Pop-up menus can be very useful in data entry. They can present to
a user the permissible entries for a field, thus (1) eliminating the
need for the user to remember the entries, (2) preventing invalid
entries, and (3) eliminating potential typing errors.

> **Definition.** A **pop-up menu** is a menu that is associated
> with a particular object on a display, for example, a pop-
> up menu listing acceptable command options close to the
> immediate work area. This is particularly useful for large
> displays, where the work site may be relatively removed
> for the menu bar.

□ **8.4.1.4.1 When to use.** Pop-up menus should be used only if it is
critical to the application that users be able to access functions
without moving the pointing device. They should not be the only
method for accessing operations, since the operations are hidden
from view, requiring users to remember where they are and how to
access them (same as paragraphs 8.1.11.6.1 and 8.2.1.8.5).

■ **8.4.1.4.2 Pop-up menu location.** A pop-up menu shall appear in a
location that is coordinated with the location of the pointer (same
as paragraph 8.1.11.6.2).

□ **8.4.1.4.3 Selection highlighting.** If an option in a pop-up menu
remains on display after it has been selected, it should remain
highlighted (same as paragraph 8.1.11.6.3).

> **Explanation.** This method is preferred to holding the
> button down while moving the cursor and releasing it to
> make a selection. The deliberate click method is less
> prone to error.

8.4.1.5 Format

□ **8.4.1.5.1 Presentation of options.** The options in a menu should be presented in a single vertical column, aligned and left-justified (see also paragraph 8.1.11.2.1).

■ **8.4.1.5.2 Consistent menus and options.** If the same menu or option appears in different displays within an application, it shall be consistent in wording and ordering (same as paragraph 8.1.11.2.2).

■ **8.4.1.5.3 Logical grouping of menu options.** If applicable, the options in a menu shall be presented in logical groups (same as paragraph 8.1.11.2.3).

■ **8.4.1.5.4 Ordering groups of options.** Groups of options in a menu shall be ordered logically. If there is no apparent logical ordering, the groups shall be ordered by their importance or expected frequency of use (same as paragraph 8.1.11.2.4).

■ **8.4.1.5.5 Ordering options within a group or menu.** If a group of options or a menu contains a small number of options, the options shall be ordered by logical sequence or frequency of use. If a group or menu contains a very large number of options, the options shall be ordered alphabetically (same as paragraph 8.1.11.2.5).

8.4.1.6 Selecting options

■ **8.4.1.6.1 Equivalence of input devices.** The system or application shall provide a user the ability to use any of the input devices available to select a menu option. For example, if a user has both a pointing device and a keyboard available, he or she shall be able to use either to select an option (same as paragraph 8.1.11.7.1).

□ **8.4.1.6.2 Menu selection by pointing.** If menu selection is the primary interactive method, and especially if selections are made from extensive lists of options, selection by pointing device should be provided (same as paragraph 8.1.11.7.6).

□ **8.4.1.6.3 Method of selecting by pointing.** The method for selecting an option by pointing should be that of moving the cursor onto the desired option and clicking the "select" button on the pointing device.

> **Explanation.** This method is preferred to holding the button down while moving the cursor and releasing it to make a selection. The deliberate click method is less prone to error.

■ **8.4.1.6.4 Initial cursor position for pointing devices.** If a user must select among displayed options using a pointing device, the cursor shall be placed on the default option when the display appears (same as paragraphs 8.1.6.10 and 8.1.11.7.2).

■ **8.4.1.6.5 Initial cursor position for keyboards.** If a user must
select among displayed options using a keyboard, the cursor shall be
placed on the default option in the control entry area (with that
control entry area having implicit input focus) when the display
appears (see "discussion" in paragraph 8.3.4.3.1) (same as
paragraphs 8.1.6.11 and 8.1.11.7.3).

■ **8.4.1.6.6 Feedback for menu selection.** If no computer response is
immediately observable when a user selects an option, the
software shall provide some other acknowledgment of the
selection. For example, the software might display a watch,
hourglass, or a message stating the delay remaining or completed
(same as paragraph 8.2.11.7.4).

□ **8.4.1.6.7 Number of selections per menu.** A user should be
allowed to select only one option from a menu. If the menu is
divided into groups a user should be able to select only one
option from each group, although users may be able to select
multiple files from a menu (same as paragraph 8.1.11.7.9).

8.4.2 Text

8.4.2.1 General

■ **8.4.2.1.1 Text input area.** The system shall provide a sufficient
screen working area that permits users to enter and edit text.

■ **8.4.2.1.2 Distinctive appearance.** Text entered by a user shall be
clearly distinguishable from system supplied text that also
appears on the screen.

□ **8.4.2.1.3 Contrast.** In general, text should be displayed as black
characters on a white or light background (same as 8.3.10.4.1 and
8.5.2.1.1).

□ **8.4.2.1.4 Multiple input devices.** If the system provides more
than one input device, for example, both a pointing device and a
keyboard, a user should not have to alternate frequently between
devices. One solution is to provide both devices the ability to
perform all operations.

■ **8.4.2.1.5 Multiple cursors.** Multiple cursors shall be avoided
unless needed for user tasks. If more than one cursor is provided,
each shall be easily distinguishable from the other(s), and the
status of each (active or inactive) shall be easily distinguishable.

■ **8.4.2.1.6 Cursor movement.** When entering and editing text,
users shall be able to move the cursor freely within a displayed
page to specify items for change, and to make changes directly in the
text.

□ **8.4.2.1.7 Enhanced cursor movement.** As applicable, users
should be able to move the cursor by units of character, line,
paragraph, and page.

□ **8.4.2.1.8 Frequently used text blocks.** If applicable, a system should provide users a means for storing and retrieving frequently used blocks of text, for example, distribution lists.

□ **8.4.2.1.9 Spell checker.** If an application involves extensive entry of text, it should provide an on-line spell checker. The spell checker should include abbreviations and acronyms or the ability to supplement the dictionary.

8.4.2.2 Text entry and editing

□ **8.4.2.2.1 Insert mode as default.** The default mode for text entry should be insert, not replace. That is, when a user types, the new text should be added at the insertion point, and the cursor and any existing text should move to the right.

□ **8.4.2.2.2 Action of Backspace and Delete.** The **Backspace** key should delete the character to the left of the text cursor and the **Delete** key should delete the character the cursor is on. If in the application, the cursor rests between characters (such as an I-beam), the **Delete** key will delete the character to the right of the text cursor.

□ **8.4.2.2.3 Editing operations.** Easy to use editing operations should be provided, including **Cut, Copy, Paste,** and **Undo**.

□ **8.4.2.2.4 Searching text.** A character string search capability should be provided that searches the text for a specified string and places the cursor at the first match found. The case (upper or lower) of the characters should be ignored unless specified otherwise by the user.

□ **8.4.2.2.5 Global search and replace.** A global search and replace capability should be provided. For example, a user should be able to command the system to locate all occurrences of the string "respond" and replace them with the string "response."

□ **8.4.2.2.6 Editing units of text** Users should be able to specify units of text for the editing operations. The units should include characters, words, lines, paragraphs, and pages.

□ **8.4.2.2.7 Highlighting units of text.** As appropriate, units of text that are designated for an editing operation, such as **Cut** or **Copy,** should be highlighted or indicated in some other way.

8.4.2.3 Formatting

□ **8.4.2.3.1 Text format.** The system should provide a default format for standard text input. If it also provides users the ability to define their own formats, it should include a means for them to store those formats for future use.

□ **8.4.2.3.2 Page formatting.** The system should provide users an easy means for specifying page formats. Formatting should include margins and tabs.

□ **8.4.2.3.3 Line breaks.** The system should provide automatic line breaks and automatic word-wrap when text reaches the right margin. The system should also provide for user-specified line breaks.

□ **8.4.2.3.4 Justification of text.** Unless otherwise specified by a user, text should be left-justified (ragged right edge) with consistent spacing between words as it is entered. Right-, center-, and full-justification should be provided as user options.

□ **8.4.2.3.5 Hyphenation.** The system should provide automatic hyphenation of words at a user's request. The default mode should be no hyphenation.

□ **8.4.2.3.6 Page breaks.** The system should provide automatic page breaks and user-specified page breaks. Users should be able to specify a minimum number of lines of a paragraph that will appear at the bottom or top of a page ("widow-orphan protection").

□ **8.4.2.3.7 Page numbering.** Automatically incremented page numbering should be provided. By default, page numbering should begin with one, but users should be able to override the default by specifying a beginning page number.

8.4.2.4 Text cursor in windows

■ **8.4.2.4.1 Text cursor.** The text cursor shall be an I-beam in insert mode and a box over a character in replace mode. The text cursor shall flash at a rate between 2 and 5 Hz. If the text object containing the text cursor loses input focus, the cursor shall stop flashing. If the text object regains input focus, the cursor shall return to normal brightness and resume flashing (same as paragraph 8.3.8.1).

> **Discussion. Input focus** means that the indicated location, window, or object in the text field is currently "active" and, unless the user changes this active state, that will be the object or location that will be acted upon by the next text editing or entry transaction.

■ **8.4.2.4.2 Text cursor location.** When a window first receives input focus, the text cursor shall be placed in the text area where typing is most likely to occur. If the cursor disappears from view when its window loses focus, the cursor shall reappear at the same location when the window regains focus (same as paragraph 8.3.8.2).

■ **8.4.2.4.3 Moving the text cursor.** Users shall be able to move the text cursor within and among text entry areas using both the pointing device and the keyboard. If a user moves the pointer

into a text entry area and clicks the appropriate button, the text cursor shall appear at the pointer location (same as paragraph 8.3.8.3).

▢ **8.4.2.4.4 Control entries distinguishable from text.** If applicable, control entries that are displayed in text (for example, paragraph indentation symbols and printer commands, such as begin and end underline) should be distinguishable from the main text.

8.4.3 Forms

Form filling as a means of data entry is especially appropriate if some flexibility is needed (such as the inclusion of optional as well as required items), if users will have moderate training, or if computer response might be slow.

8.4.3.1 General

■ **8.4.3.1.1 Title.** Each form shall have a title. The title shall appear at the top of the form.

■ **8.4.3.1.2 Consistency.** Forms, labels, fields, messages, and instructions that appear on different displays shall be as consistent as possible within an application and among related applications.

■ **8.4.3.1.3 Field help.** Help shall be provided for fields.

> **Discussion.** Some help might be provided automatically when the cursor arrives in a field, such as an explanatory message or a menu of acceptable entries. Other ways context-sensitive help might be provided include: (1) providing a **Help** operation that provides help on the field that contains the cursor, and (2) providing help on the field when a user moves the pointer onto the field label and clicks the appropriate button.

8.4.3.2 Fields

■ **8.4.3.2.1 Appearance.** Fields shall have a distinctive appearance and distinct limits, for example, a series of underscores, or a rectangle, perhaps in inverse video, that clearly distinguish fields from each other and from other objects and information on the screen.

▢ **8.4.3.2.2 Field length.** Data entry fields should be of fixed length, even if the entries may be of variable length. If useful to the user, a field should give a cue as to its length, for example by using separated underscores (_ _ _ _ _).

■ **8.4.3.2.3 Entry does not overwrite field delineators.** Fields shall not be designated by characters that are overwritten as a user enters data.

■ **8.4.3.2.4 Unfilled portion of field.** If a field accepts variable length entries, users shall not have to remove or fill any unneeded portion.

■ **8.4.3.2.5 Required fields.** If a form has one or more required
fields, the user shall have to make an entry in each required field to
be able to complete the form in its intended way. The save
button shall be displayed as unavailable until all of the required
fields have been filled (see also paragraphs 8.4.3.8.3 and
8.4.3.8.4).

>   **Examples.** A user might be given an error message if he or
>   she tries to leave a required field without making an entry,
>   or a user might be given an error message if he or she tries
>   to **Save** a form without making an entry in all required
>   fields.

■ **8.4.3.2.6 Optional fields distinct from required fields.** If a form
has both optional and required fields, the two types of fields shall be
easily distinguishable.

>   **Examples.** One way to do this would be to use different
>   label terminators for the two types of fields, for example, the
>   labels of optional fields might be followed by a colon (:),
>   and the labels of required fields might be followed by
>   a slash (/) (see paragraph 8.4.3.3.5). Another way to do
>   this would be to use different appearances for the fields
>   themselves, for example, a required field might appear as
>   underscores (_____), and an optional field as a row of
>   dots (......).

■ **8.4.3.2.7 Intrafield separators.** If possible, fields provided for
data that include separators or some sort of formatting, such as
slashes separating the month, day, and year in dates, or a decimal
point separating dollars and cents, shall include the separators or
formatting as part of the field.

>   **Examples.** A field for a date might appear:

>   DATE: __/__/__ .

>   A field for a telephone number might appear:

>   TELEPHONE NUMBER: (___) ___-____ .

8.4.3.3 Field labels

■ **8.4.3.3.1 Field labels.** Every data field shall have a label that
uniquely identifies the field.

>   **Discussion.** A single label is sufficient for a series of
>   fields of the same type arrayed in a row or column.

▢ **8.4.3.3.2 Descriptive labels.** A label should specify or suggest
the entry that goes into the field. Numbers and other arbitrary
codes should not be used as field labels.

>   **Discussion.** Complete words are preferred as labels, but
>   predefined terms, codes, and abbreviations may be
>   acceptable.

☐ **8.4.3.3.3 Terms used in labels.** Labels for data fields should be composed of terms that are familiar to the user, relevant to the topic of the form, and easily understood by a typical user.

■ **8.4.3.3.4 Labels distinct from other information.** Labels shall be distinct from data entries and from other information on the screen, for example, by differing in font or size.

■ **8.4.3.3.5 Label terminator.** Field labels shall terminate with a special symbol that designates the end of the label and the beginning of the field. A colon (:) is frequently used for this purpose. If the label is to the left of the field, the terminator shall be followed by a blank space that separates it from the beginning of the field.

■ **8.4.3.3.6 Consistent location.** Labels shall be located consistently with respect to their fields.

> **Discussion.** The preferred location for a label is to the left of or above its field. If a form contains both single label-field pairs and arrays (rows or columns) of fields with a single label, the location of labels for the single label-field pairs may be different from the labels for the arrays of fields.

■ **8.4.3.3.7 Unit of measurement.** If a field entry involves a unit of measurement, the unit shall be included as part of the label or field.

> **Examples.**
>
> COST: $_____.__
>
> LENGTH (ft): _____

■ **8.4.3.3.8 Alternative units.** If measurements might be in different units, for example, inches or millimeters, users shall not have to transform them at the time of data entry.

> **Discussion.** This problem might be solved by providing a field for each unit of measurement, and the user selects the correct field. Another solution might be to have one field for the quantity and another field for the unit of measurement.

■ **8.4.3.3.9 Labels not editable.** Field labels shall not be editable by users, at least not while they are in form-filling mode.

8.4.3.4 Layout

■ **8.4.3.4.1 Correspondence between screen and document.** If users will transfer data from hard copy documents, the screen layout shall correspond to the hard copy in the order and grouping of data items. It is desirable that the displayed form look as much like the source document as possible.

■ **8.4.3.4.2 Layout with no source document.** If input is not from source documents or hard copy forms, data fields shall be ordered and grouped logically, using sequence, frequency of use, importance, and functional associations as organizing principles.

□ **8.4.3.4.3 Multipage forms.** If a form is too large to fit in the available screen area, it should be broken into "pages," and each page should be labeled with its number and the total number, for example, **Page 1 of 3.**

8.4.3.5 Navigation

■ **8.4.3.5.1 Initial cursor position.** When a form first appears, the cursor shall be placed automatically in the first position of the first field.

■ **8.4.3.5.2 Easy cursor movement.** The system shall provide one or more easy ways to move the cursor among fields. If the primary means of entering data in fields is the keyboard, the cursor movement methods shall include keyboard keys such as the **Tab** key(s) and the arrow keys. If a pointing device is available, a user shall be able to move the cursor to any field by moving the pointer into the field and clicking the appropriate button. If both a keyboard and pointing device is available, cursor movement shall be allowed using either device.

□ **8.4.3.5.3 No automatic movement.** The cursor should not be moved automatically among fields; movement should occur only upon explicit user action, such as pressing the **Tab** key.

> **Exception.** There may be cases in which automatic movement is desirable. For example, if skilled users enter numerous entries of fixed length, it may be preferable to move the cursor automatically to the next field when the current field is filled. The danger is that a missed or extra character may result in erroneous entries in many fields before the user notices.

■ **8.4.3.5.4 Navigation only to fields.** In general, a user shall be able to move the cursor only into fields and onto control objects on the screen; that is, a user shall not be able to move the cursor onto labels or other nondata-entry areas on the screen.

■ **8.4.3.5.5 Protected fields.** If a form has protected fields, a user shall not be able to move the cursor into a protected field.

> **Explanation.** A field might be protected from some users and not from others. Other fields might be reserved for the display of computed values.

□ **8.4.3.5.6 Moving to "next" and "previous" fields.** If the fields in a form will be traversed sequentially, a user should be able to move the cursor to the "next" field by pressing the **Tab** key, and to the "previous" field by pressing the **Shift** and **Tab** keys simultaneously.

Discussion. This sort of movement requires a predefined "path" through a form that specifies which field is "next" and which is "previous." Presumably such a path will traverse each field once and only once in a systematic way, for example, from left to right and top to bottom.

□ **8.4.3.5.7 Navigation with a pointer.** If fields may not necessarily be traversed in a set order, a pointing device, in addition to keyboard, should also be available for selecting fields.

8.4.3.6 Defaults

□ **8.4.3.6.1 When to use.** If a form is expected to have the same entry in a particular field most of the time, that entry should appear in that field as a default entry when the form first appears.

■ **8.4.3.6.2 Displaying default values.** A field that has a default value shall have that value appear in the field automatically when the form appears.

■ **8.4.3.6.3 Replacing default values in fields.** If an entry is normally made in a field by typing, a user shall be able to replace a default value by moving the cursor into the field and typing. The default value shall disappear immediately after the first keystroke. This action shall not affect the default value itself; that is, the next time the form appears, the same default value shall appear in the field.

Exception. An exception to this rule is when an application permits a user to select whether he or she wants the application to retain the last entry or a previous default value as the current default setting.

8.4.3.7 Data entry and editing

■ **8.4.3.7.1 Unfilled spaces.** Users shall not have to move a space at a time over unfilled spaces in variable length fields.

■ **8.4.3.7.2 Leading and trailing zeros.** A user shall not have to enter leading or trailing zeros to fill a field.

■ **8.4.3.7.3 Justification of entries.** If a user makes an entry that does not fill a variable length field, the system shall justify the entry automatically when the cursor leaves the field. Unless otherwise required by processing or display requirements, justification shall be as follows:

a.   Alphanumeric input shall be left justified.

b.   Integer numerical data shall be right justified.

c.   Decimal numerical data shall be decimal point justified.

■ **8.4.3.7.4 Interrupt capabilities.** Users shall have the ability to use **Backup** (see paragraph 8.1.4.3), **Cancel** (see paragraph 8.1.4.4), and **Restart** (see paragraph 8.1.4.8) actions to edit a form at any time prior to the final "completion" action.

■ **8.4.3.7.5 Editing entries.** Users shall be able to move the cursor to any unprotected field and change any entry prior to taking a final "completion" action.

■ **8.4.3.7.6 Explicit "completion" action.** A form shall not be removed from display until the user takes an explicit "completion" action, such as pressing the **Enter** key.

8.4.3.8 Error management

■ **8.4.3.8.1 Easy error correction.** Users shall be able to correct errors easily on a character-by-character and field-by-field basis.

■ **8.4.3.8.2 Unacceptable entries.** If a field has a set or range of acceptable values, and a user enters an unacceptable value, the system shall either:

a.   provide an error message when the user tries to leave the field and not move the cursor from the field, or

b.   allow the user to continue moving through the form and, when the user tries to perform the "completion" action, provide an error message and move the cursor to the field in error.

■ **8.4.3.8.3 Omitted fields.** If a user fails to make an entry in a required field, the system shall either:

a.   provide an error message when the user tries to leave the field and not move the cursor from the field, or

b.   allow the user to continue moving through the form and, when the user tries to perform the "completion" action, provide an error message and move the cursor to the field in error.

□ **8.4.3.8.4 Deliberate omissions.** If applicable, a system or application should provide a special symbol that a user can enter in a required field; this symbol will allow the user to defer the required entry and continue with the remainder of the form.

8.4.4 Direct manipulation

In a graphical user interface, a major type of interactive dialog is direct manipulation. In a direct manipulation dialog, the user controls the interface with the computer by acting directly on "objects" on the display screen. An object may be an icon, menu option, symbol, button, or dialog box.

□ **8.4.4.1 Drag transfer.** If a system provides direct manipulation, a user should be able to move and copy data and objects by first marking the data or object, if necessary, then placing the pointer

on it, holding down the appropriate button on the pointing device, and dragging it to the desired location.

□ **8.4.4.2 Data and object transfer.** If a system provides direct manipulation, a user should be able to move and copy data and objects by first marking the data or object, if necessary, then transferring the data or object to a temporary storage, and then transferring the data or object from the temporary storage to its new location.

8.4.5 Tables

□ **8.4.5.1 When to use.** If sets of data must be entered sequentially or if data are keyed row by row, a tabular format should be used.

■ **8.4.5.2 Labels.** Each row and column shall be uniquely and informatively labeled, and the labels shall be distinct from the data cells.

■ **8.4.5.3 Leading and trailing zeros.** Users shall not have to type leading zeros (before numbers to the left of the decimal point) or trailing zeros (following numbers to the right of the decimal point).

■ **8.4.5.4 Automatic justification.** Data typed into a cell of a table shall be justified automatically when the user moves the cursor to the next cell. Justification shall be as follows (see also paragraph 8.4.3.7.3):

a. Alphanumeric input shall be left justified.

b. Integer numerical data shall be right justified.

c. Decimal numerical data shall be decimal point justified.

■ **8.4.5.5 Navigation with the Tab key.** The **Tab** key shall move the cursor to the first position of the next cell to the right of its current position, or if the current position is in the last cell in a row, it shall move the cursor to the first position of the first cell in the next row. Similarly, pressing **Shift** and **Tab** simultaneously shall move the cursor to the first position in the next cell to the left of the current position, or if the current position is in the first cell in a row, it shall move the cursor to the first position in the last cell in the preceding row.

■ **8.4.5.6 Navigating with a pointing device.** If a pointing device is available, a user shall be able to move the cursor to any cell by moving the pointer into the cell and clicking the appropriate button.

■ **8.4.5.7 Large tables.** If a table is too large to fit in the available display area, as much of the top left portion shall be displayed as will fit when it is first displayed, and appropriate scroll bars or similar mechanisms shall be provided. Scroll bars may be provided on the right or left side, and on the bottom or top.

■ **8.4.5.8 Labels in scrolling tables.** When a user scrolls a large table, the row or column labels that remain relevant shall not scroll, but shall remain in place. For example, if the rows scroll up or down, the column labels shall remain in place.

8.4.6 Entry of graphics

8.4.6.1 General

□ **8.4.6.1.1 Pointing device.** The system should provide a pointing device for entering and manipulating graphic data. The pointing device should also be capable of system control.

□ **8.4.6.1.2 Graphics cursor.** The cursor for creating graphics displays should be (1) distinctive, (2) easy to position, and (3) have a point that can be used to select and manipulate small graphic objects. Some good and bad examples of graphics cursors are given in exhibit 8.4.6.1.2.

Exhibit 8.4.6.1.2 Examples of better and worse graphics cursors

Pointing cursors too vague or obscuring

More precise pointing possible
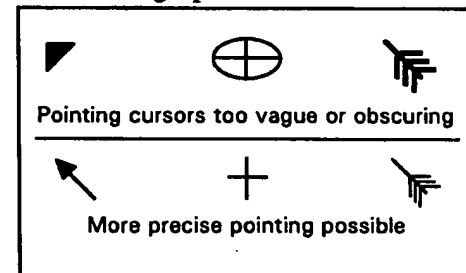
□ **8.4.6.1.3 Graphics cursor operation.** A graphics cursor operation should have a movement (pointing) component and an activation component. The movement component should position the cursor, and the activation component should activate the position to manipulate a display element, for example, selecting an object to move or drawing a line.

□ **8.4.6.1.4 Validation on input.** To the extent possible, the system should validate graphic data as it is created. For example, the system should provide a message if a given value is outside the standard range.

■ **8.4.6.1.5 Saving and retrieving graphic data.** An easy means shall be provided for saving and retrieving graphic data. Users shall be able to specify names for storing graphic data files and be able to view lists of these stored files.

8.4.6.2 Graphics entry and editing

□ **8.4.6.2.1 Drawing lines.** The system should draw lines between user specified points and should support the drawing of rectangles, circles, arcs, ovals, and other figures.

□ **8.4.6.2.2 Constraining lines.** Users should be able to constrain lines to be exactly vertical or horizontal. They should also be

able to specify that a line is perpendicular or parallel to another line.

□ **8.4.6.2.3 Alignment grid.** The system should provide the capability of aligning objects on an invisible rule or grid structure at a user's request. Users should be able to specify grid intervals.

□ **8.4.6.2.4 Alternate drawing methods.** If required by the task, alternate methods should be provided for drawing objects. For example, a circle might be drawn by specifying a center and a radius or diameter, or by specifying the size and location of an enclosing square.

□ **8.4.6.2.5 Closure.** Users should be able to select automatic figure completion, that is, automatic closure of polygons. If separately drawn lines must connect at terminal points, the system should automatically make the connections.

□ **8.4.6.2.6 Displaying attributes.** If desired by the user, object attributes should be displayed as selected. When displayed, they should not be represented as appended codes or by some other means.

□ **8.4.6.2.7 Colors and patterns.** Users should be able to fill enclosed areas with colors or patterns.

□ **8.4.6.2.8 Selectable elements and attributes.** Users should be able to select and edit display elements (such as lines) and their attributes (such as thickness) by pointing to and selecting from displayed examples.

□ **8.4.6.2.9 Manipulating objects.** Users should be able to copy, rotate, and reverse (produce mirror images) objects both horizontally and vertically.

□ **8.4.6.2.10 Editing objects.** User-selectable objects should be easily repositioned, duplicated, and deleted.

□ **8.4.6.2.11 Scaling objects.** Users should be able to enlarge and reduce the size of objects.

□ **8.4.6.2.12 Zoom capability.** A zoom capability should be provided to enlarge critical display areas.

□ **8.4.6.2.13 Overlapping objects.** If it is desired by the user, the system should automatically merge objects and assign them precedence. For example, if two objects overlap, the system should obscure the overlapped portion of one of the objects.

□ **8.4.6.2.14 Grouping objects.** The system should provide a means to group separate objects into a single grouped object that can then be treated as a single object.

8.4.6.3 User aids

☐ **8.4.6.3.1 Entering data for plotting.** When complex graphic data must be entered quickly, computer aids should be provided. For example, when plotting data within Cartesian coordinates, the system should automatically draw lines between the specified points of a function.

☐ **8.4.6.3.2 Plotting stored data.** The system should support automatic plotting of stored data.

☐ **8.4.6.3.3 Scaling graphic data.** The system should provide for automatic scaling of graphic data, and users should be able to modify system-generated scales.

☐ **8.4.6.3.4 Emergence of drawn objects.** Objects should emerge as they are being drawn. For example, when a user draws a line by moving a pen across a graphics tablet, the line displayed should emerge as the pen moves from the start point, increasing or decreasing in length and slope as the pen moves across and around the tablet.

8.4.7 Data validation

☐ **8.4.7.1 Format and content.** When possible, the system should automatically check data for format and content. For example, a date entered as February 31 should result in a content error message.

☐ **8.4.7.2 Valid data.** Valid data entries should be accepted and processed without any further user involvement, for example, if data pass validation tests, the user should not be prompted for a confirmation.

☐ **8.4.7.3 Invalid data.** Data and command entries that do not meet validation testing should result in a message asking for correction or confirmation.

☐ **8.4.7.4 Probable errors.** If validation testing detects a probable error, an error message should be displayed at the completion of the data entry, without interrupting an ongoing transaction.

8.5 Data display

Data display refers to the computer's presentation of information to the user. The emphasis in this section is on information presented on visual display terminals, but it also includes rules about auditory signals and displays.

8.5.1 General

☐ **8.5.1.1 Independence.** The content of each screen should stand on its own; users should not have to refer to a previous screen or remember essential information. For example, if the same information is needed in a series of screens, the system might

prompt the user to enter the information on the first screen and then automatically enter the information on subsequent screens.

■ **8.5.1.2 Consistent with user expectations.** Data shall be displayed consistently, using standards and conventions familiar to users.

■ **8.5.1.3 Consistent within applications.** Data display shall be consistent in word choice, format, and basic style throughout an application and related applications.

□ **8.5.1.4 Whole data sets.** Whenever possible, users should be able to see the whole data set of interest, for example, an entire page, map, or graphic.

□ **8.5.1.5 Information density.** Information density should be minimized in displays used for critical task sequences. For critical information, a minimum of one character space should be left blank vertically above and below critical information, and a minimum of two character spaces should be left blank to the left and to the right of the critical information.

■ **8.5.1.6 Usable, essential data for a transaction.** The data needed for a transaction shall be displayed in a directly usable form, and only essential data shall be displayed.

□ **8.5.1.7 User control.** Users should be able to control the amount, format, and complexity of displayed data, as necessary to meet task requirements.

□ **8.5.1.8 Paper copy.** Users should be able to obtain a paper copy of the exact contents of an alphanumeric or graphic display in systems in which mass storage is limited, mass stored data can be lost by power interruption, or record keeping is required.

□ **8.5.1.9 Date and time information.** If task performance requires or implies the need to assess the timeliness of information, the display should include time and date information associated with the data.

□ **8.5.1.10 Familiar wording.** The wording of displayed data and labels should use familiar terms and the task-oriented language of the users; unfamiliar terms and language should be avoided.

■ **8.5.1.11 Display formats.** The different elements of display formats shall be distinctive within a display, and consistent across displays.

□ **8.5.1.12 Blank space.** Blank space should be used to structure a display.

□ **8.5.1.13 Grouped information.** Groups of data items should be separated by blank space, lines, color coding, or other visually distinctive means.

□ **8.5.1.14 Reserved area.** If the user's task requires frequent referral to or use of status or error messages, prompts, or command entries, those elements should be displayed in a reserved area at the bottom of the display.

□ **8.5.1.15 Layout for comparisons.** If users must analyze sets of data for similarities, differences, or trends, displays should be formatted so that the data are grouped and aligned to facilitate these analyses.

□ **8.5.1.16 Character-by-character comparisons.** If data fields are to be compared character-by-character, the fields should be vertically aligned.

■ **8.5.1.17 Arranging data.** If applicable, data shall be arranged by sequence, function, importance, frequency of use, or by other means such as chronologically or alphabetically.

□ **8.5.1.18 Context.** Context should be provided for displayed data. For example, if a user is changing parameters for a facility, relevant information concerning that facility should be displayed.

□ **8.5.1.19 Multipage displays.** If a data set contains too much data for presentation in a single display, the data should be partitioned into separately displayable pages.

□ **8.5.1.20 Partitioning data among pages.** Related data should appear on the same page. Relations among data sets should appear in an integrated display rather than being partitioned into separate pages.

□ **8.5.1.21 Labeling pages.** Each page in a multipage data set should be labeled to show its relation to the others. For example, the first page of a three-page set might be labeled **Page 1 of 3.**

8.5.2 Text

8.5.2.1 General

■ **8.5.2.1.1 Consistent wording and structure.** The wording and grammatical structure of displayed data and labels shall be consistent throughout an application and related applications.

□ **8.5.2.1.2 Contrast.** In general, text should be displayed as black characters on a white or light background (same as 8.3.10.4.1 and 8.4.2.1.2).

8.5.2.2 Labeling

□ **8.5.2.2.1 Distinct, unique, descriptive labels.** Each data group, message, or display should contain a distinct, unique, descriptive, and consistently worded title or label.

□ **8.5.2.2.2 Alphanumeric labels.** The labels of screens should be alphanumeric. If they are not complete words, labels should be

abbreviations that are short enough (3 to 7 characters) or meaningful enough to be learned and remembered easily.

□ **8.5.2.2.3 Consistency.** Label locations and formats should be consistent.

□ **8.5.2.2.4 Spacing.** At least one blank line should separate a title from the body of a display.

8.5.3 Forms

□ **8.5.3.1 Distinctive fields.** Data fields should be visually distinguishable from other displayed information (see also section 8.4.2).

8.5.4 Coding

8.5.4.1 General

□ **8.5.4.1.1 Meaningful codes.** If codes are used, they should be meaningful rather than arbitrary. For example, "male" and "female" might be coded "M" and "F," rather than "1" and "2."

□ **8.5.4.1.2 When to use.** If coding is used, it should (1) differentiate items of information, (2) call a user's attention to changes in the state of a system, or (3) indicate important, hazardous, or critical information that requires user action.

□ **8.5.4.1.3 Coding data categories.** Categories of data should be coded if a user must distinguish the data included in the categories rapidly and if the data items are distributed in an irregular way on the display.

■ **8.5.4.1.4 Consistent coding.** Coding shall be consistent throughout an application and related applications.

□ **8.5.4.1.5 Special codes.** Codes that are assigned a special meaning in a display should be defined at the bottom of the display.

8.5.4.2 Alphanumeric coding

□ **8.5.4.2.1 Supplemental use only.** Alphanumeric coding should not be used as the sole means to call attention to important or critical information. It may be used to supplement other coding schemes.

□ **8.5.4.2.2 Case of letters.** Alphanumeric codes should use either upper case letters or lower case letters consistently; they should not use mixed cases.

□ **8.5.4.2.3 Mixed letter and number codes.** If short codes contain both letters and numbers, the letters should be grouped together

and the numbers should be grouped together. For example, the code HW5 might be used rather than the code H5W.

□ **8.5.4.2.4 Length of codes.** Arbitrary alphanumeric codes that are intended to be recalled by users should have no more than five characters and should be the same length.

8.5.4.3 Auditory coding   Coded auditory signals are appropriate to: (1) alert users to critical conditions or operations, (2) supplement visual signals, (3) present information in situations in which visual presentation is not feasible, and (4) provide feedback for control actuation, data entry, or the completion of timing cycles and sequences.

■ **8.5.4.3.1 Acknowledging auditory signals.** A simple, consistent means of acknowledging auditory signals shall be provided. If the signal is noncritical, the acknowledgement action shall also turn the signal off.

■ **8.5.4.3.2 User control.** Users shall be able to turn off noncritical auditory signals.

□ **8.5.4.3.3 Delayed computer response.** If the computer response to a user request is greater than 15 seconds, the computer should provide an auditory signal when it responds.

□ **8.5.4.3.4 Nature of auditory signals.** Different auditory signals should be easily distinguishable, for example, by varying in frequency, modulation, or both. Auditory signals should be intermittent rather than continuous.

□ **8.5.4.3.5 Environmental compatibility.** The intensity, duration, and source location of an auditory signal should be compatible with the acoustic environment of the intended receiver as well as with the requirements of other personnel within acoustic range of the signal.

8.5.4.4 Brightness intensity coding

■ **8.5.4.4.1 Consistent meaning.** Brightness coding shall have a single meaning throughout an application and related applications; for example, two brightness levels might mean **ON** and **OFF**, or **FAST** and **SLOW**, or **STANDBY** and **RUN**, but only one of the three.

■ **8.5.4.4.2 Number of levels.** The number of brightness intensity levels used as codes shall not exceed three.

■ **8.5.4.4.3 Brightness ratios.** Each level of brightness shall be separated from an adjacent level by a 2:1 ratio.

8.5.4.5 Color coding   Color coding can be helpful in differentiating classes of information in complex, dense, and critical displays. The color of the figure, background, and surrounding needs to be considered in order to provide the appropriate contrast and emphasis to the color coding.

■ **8.5.4.5.1 Reserved meanings.** Color coding shall conform to the following reserved meanings:

a. Red shall indicate conditions such as "no-go," "error," "failure," or "malfunction."

b. Flashing red shall be used only to indicate emergency conditions requiring immediate user action to avert personnel injury or equipment damage.

c. Yellow shall indicate marginal conditions, alert users to situations where caution or rechecking is necessary, or notify users of an unexpected delay.

d. Green shall indicate that a monitored process or unit of equipment is within tolerance, that a condition is satisfactory, or that it is all right to proceed with an operation or transaction.

e. White shall indicate alternative functions or system conditions that do not have operability or safety implications.

f. Blue shall be used only as an advisory color.

Discussion. The use of colors to indicate primary meanings is also dependent on the color appearing against an appropriately contrasting background. For instance, white or light gray are appropriate for black text.

■ **8.5.4.5.2 Color coding data categories.** If color is used to identify data categories, its use shall not conflict with other color coding conventions, or with those in paragraph 8.5.4.5.1.

□ **8.5.4.5.3 Redundant use.** Color coding should not be used alone; it should be redundant to some other means of coding, such as symbols or size.

■ **8.5.4.5.4 Use of color.** Colors shall be easily discriminable, and color shall be used conservatively and consistently, with each color representing only one category of displayed data.

□ **8.5.4.5.5 Drawing attention.** Brighter or more saturated colors should be used to draw a user's attention to critical data.

8.5.4.6 Flash coding

■ **8.5.4.6.1 Limited use.** Flash coding shall be used only to indicate an urgent need for a user's attention.

■ **8.5.4.6.2 Flashing rate.** The rate of flashing shall be in the range of three to five flashes per second, with equal on and off durations (see also paragraph 8.5.4.6.3).

■ **8.5.4.6.3 Second flashing rate.** The number of flashing rates shall not exceed two. If two rates are used, one shall be slower than two flashes per second.

■ **8.5.4.6.4 Displayed objects.** If a displayed object is to be flash coded, a flashing symbol adjacent to the object shall be used rather than flashing of the object itself.

□ **8.5.4.6.5 Flash acknowledgement.** If flash coding is used, users should have a means of acknowledging the flashing. If appropriate, this acknowledgement should automatically stop the flashing.

8.5.4.7 Line coding   Lines can be coded by such attributes as width or thickness, color, and pattern (that is, solid, dashed, dotted, and so on).

□ **8.5.4.7.1 Length.** Quantities, such as velocity or distance, should be coded by line length.

□ **8.5.4.7.2 Direction.** Spatial categorization in two dimensions, for example, an aircraft bearing, should be coded by line direction.

□ **8.5.4.7.3 Number of coded lines.** The number of different lines used as codes should not exceed six.

8.5.4.8 Symbol coding

□ **8.5.4.8.1 Design of symbols.** To the extent possible, a symbol should be: (1) an analog of the object it represents, (2) in general use and well known to the users, or (3) based on established standards or conventional meanings.

■ **8.5.4.8.2 Special symbols.** If special symbols, such as asterisks or arrows, are used, they shall be used consistently and with unique meanings throughout an application and related applications.

8.5.4.9 Shape coding

■ **8.5.4.9.1 Number of shape codes.** The number of different shapes used as codes shall not exceed 15.

8.5.4.10 Size coding

■ **8.5.4.10.1 Number of sizes.** The number of different sizes used as codes shall not exceed three.

8.5.5 Display of graphics   The goal of graphic presentation is to communicate information clearly and unambiguously, and to facilitate the detection of relationships among variables, comparisons among data sets, and the detection of trends in the data. This section contains criteria and guidelines for pictures and diagrams, as well as material relating to the construction of graphs and charts.

8.5.5.1 General

■ **8.5.5.1.1 Complex formats.** Complex formats and embellishments that do not convey useful information shall be avoided.

□ **8.5.5.1.2 Robustness.** Graphics should be designed to remain useful when reproduced or reduced in size.

■ **8.5.5.1.3 Appropriateness of format.** The format shall be appropriate to the user's level of training and experience.

□ **8.5.5.1.4 Only needed data.** Only the data needed by the user should be presented.

□ **8.5.5.1.5 User selection of style.** If appropriate, users should be able to select alternative styles of presentation.

□ **8.5.5.1.6 Value display.** If appropriate, users should be able to select a data point on a graph and obtain a display of the associated value or values.

> **Discussion.** Users might also be given the option of choosing between tabular and graphical displays.

■ **8.5.5.1.7 Consistency.** Graphics shall be consistent in design, format, and labeling throughout an application and related applications.

■ **8.5.5.1.8 Labels.** Displayed graphics shall be clearly labeled.

8.5.5.2 Display of critical data

■ **8.5.5.2.1 Reference values.** If users are required to make comparative evaluations against reference values, the reference values shall be displayed.

□ **8.5.5.2.2 Displaying data values with graphics.** If precise readings of values are required, the actual data values should be displayed in addition to the plotted data.

■ **8.5.5.2.3 Consistent labeling location.** If graphic data are labeled, the text shall appear in a consistent location in relation to the graphic elements.

□ **8.5.5.2.4 Supplementary text.** Supplementary text within the framework of the graph should only be used to emphasize features of data requiring user attention or to enhance user understanding. The use of supplementary text should be minimized.

8.5.5.3 Creating and editing

Computer aids such as those listed in this section need to be provided for the entry and organization of complex graphic data.

□ **8.5.5.3.1 Validation.** Data entered should be validated by the application software.

> **Discussion.** Validation might include comparison with a range or set of values, or calculated or logical relationships with other entries.

□ **8.5.5.3.2 Plotting aids.** If plotting formats are known, templates or other data entry aids should be provided.

□ **8.5.5.3.3 Plotting stored data.** The application should provide automated or aided plotting and editing of stored data.

□ **8.5.5.3.4 Automated production of scales.** The application should automatically adjust the range of scales or provide the user with automated aids for scaling graphic data.

□ **8.5.5.3.5 Line drawing.** The application should provide users automated aids for drawing straight and curvilinear lines.

□ **8.5.5.3.6 Automatic completion of polygons.** The application should provide automatic completion to users drawing polygons. That is, the application should automatically provide a line that connects the current cursor position to its starting point. A user should be able to make the provided line a permanent part of the figure.

□ **8.5.5.3.7 Joining lines.** The application should provide automated assistance in joining lines.

□ **8.5.5.3.8 Designating line segments.** Users should be able to identify and select line segments for moving and editing.

□ **8.5.5.3.9 Grid references.** The application should provide optional, adjustable grid references to aid users in aligning horizontal and vertical lines.

□ **8.5.5.3.10 User-specified rules.** Users should be able to specify rules for attributes, relationships, and design, and have the computer apply those rules automatically during the design process. For example, a user might specify that hand-drawn lines be straightened or that the angles between intersecting lines be adjusted.

□ **8.5.5.3.11 Computer aids.** The application should provide prompts and computer-aided methods for drawing figures.

□ **8.5.5.3.12 Scale changes.** The application should allow users to edit or create drawings in a large scale and then reduce them to the desired scale.

□ **8.5.5.3.13 Basic operations.** The application should allow users to resize, copy, move, rotate, and produce mirror images of objects.

□ **8.5.5.3.14 Grouping elements.** The application should allow users to select and group elements that can then be treated as a single object.

□ **8.5.5.3.15 Area fill capability.** The application should allow users to fill enclosed areas with selected attributes such as color or patterns.

□ **8.5.5.3.16 Computer models.** The application should provide models that allow a user to create a display by specifying parameters. For example, the application might have a model of a pie chart that would allow a user to create a chart by simply specifying the number and size of the segments.

8.5.5.4 Scales, labels, and coding

■ **8.5.5.4.1 Standard conventions.** Scales shall conform to the following conventions:

a. Values shall increase with distance from an origin.

b. Independent variables shall be plotted along the horizontal axis.

c. Dependent variables shall be plotted along the vertical axis.

■ **8.5.5.4.2 Consistent use of symbols.** Symbols, if used, shall be assigned unique meanings and used consistently throughout an application and related applications (see also paragraph 8.5.4.8.2).

□ **8.5.5.4.3 Color and pattern coding.** If colors or patterns are used to fill enclosed areas, they should conform to the following rules:

a. Color coding should be redundant with another form of coding.

b. If the graphic is not likely to be printed, color should be used rather than patterning.

c. If the graphic is likely to be printed, patterning should be used rather than color.

□ 8.5.5.4.4 Patterns. If patterns are used, they should be simple hatching and shading, not complex patterns that produce visual illusions of vibration or motion. Exhibit 8.5.5.4.4 illustrates acceptable and unacceptable patterns.

Exhibit 8.5.5.4.4 Examples of acceptable and unacceptable patterns
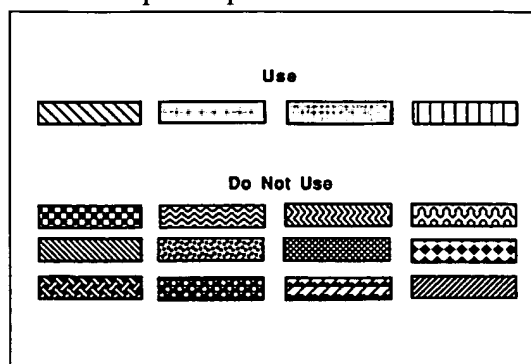
■ 8.5.5.4.5 **Breaks in axes.** If data are concentrated in a way that makes it desirable to show only a portion of an axis of a graph, the axis shall include the origin and be drawn with a break in it as illustrated in exhibit 8.5.5.4.5.

Exhibit 8.5.5.4.5 Example of axes with breaks
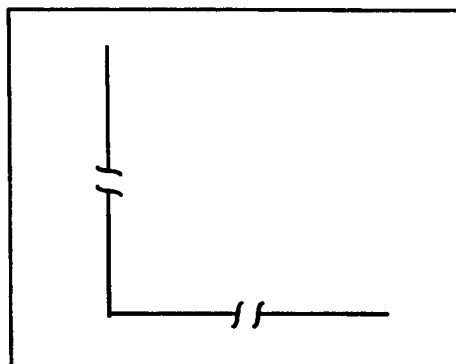
□ 8.5.5.4.6 **Duplicate axes.** If necessary to make a graph more readable, one or both of the horizontal and vertical axes should be repeated at the top or right of the graph, as appropriate.

■ 8.5.5.4.7 **Consistent formats.** If separate graphs are to be compared, or if different sets of data are to be plotted on the same graph, the formats and scales shall be identical (see also paragraphs 8.5.5.6.2 and 8.5.5.6.3).

□ 8.5.5.4.8 **Linear scales.** In general, linear scales should be used rather than other types, such as logarithmic.

> **Discussion.** Logarithmic scales may be appropriate for comparing rates of change.

□ 8.5.5.4.9 **Single scale per axis.** An axis should represent only a single scale.

■ 8.5.5.4.10 **Labeling axes.** Each axis shall have a label that describes the axis and its units of measurement. Each axis shall have tick marks corresponding to major scale divisions, and these marks shall be numbered or labeled.

□ **8.5.5.4.11 Scale divisions.** Scales should not have more than 12 major scale divisions, and each major division should not be subdivided into more than 10 parts.

■ **8.5.5.4.12 Numeric scales.** Numeric scales shall begin with zero, cover the entire range of the data, and, if applicable, the major divisions shall be labeled with decimal multiples of whole numbers.

> **Discussion.** This rule prevents the distortion or misinterpretation of data that can result if the origin is omitted or if the scale does not continuously span the data range. It also helps make valid comparisons of different graphs possible.

□ **8.5.5.4.13 Label format.** Labels should use upper and lower case sans serif fonts, and they should be oriented to permit normal left-to-right reading.

□ **8.5.5.4.14 Labeling data elements.** Labels, rather than legends or keys, should be used to identify plotted data elements. They should be located adjacent to the elements they identify, and they should be oriented to permit normal left-to-right reading.

> **Discussion.** If it is awkward to place the labels adjacent to the elements, they may be connected to the elements by arrows, lines, or other pointing conventions.

■ **8.5.5.4.15 Location of legends and keys.** If a graph requires a legend or key, the legend or key shall be located inside the rectangular bounds of the graph unless such a location would interfere with interpretation of the displayed data.

8.5.5.5 Grid lines   The addition of grid lines to graphs can be helpful to users.

> **Definition. Grid lines** are horizontal lines, vertical lines, or both, extending from the scale divisions of one or both axes of a graph, and intended to aid users in locating and reading data points.

□ **8.5.5.5.1 When to use.** Grid lines should be used only when they are necessary to help users achieve a desired level of precision. Users should have the option to easily turn grid lines on or off.

□ **8.5.5.5.2 Grid lines vs. data.** Grid lines should be easily distinguishable from data, and they should not obscure data.

□ **8.5.5.5.3 User choice.** If grid lines are provided, they should be provided in a way that gives users the option of displaying them or not.

8.5.5.6 Lines and curves

□ **8.5.5.6.1 Use of lines and curves.** Straight lines between data points or smoothed curves through the points should be used to show relationships between two variables.

□ **8.5.5.6.2 Labeling and highlighting multiple lines and curves.** If a graph contains more than one line or curve, each one should have an identifying label. If a legend is used to identify the lines, then, to the extent possible, they should appear in the legend in the same order they appear in the graph. If one curve or line is critical, that one should be highlighted.

> **Discussion.** The preferred location for labeling a line or curve is adjacent to it, but, if the spacing of the lines or curves makes this difficult, it is acceptable to use a legend.

■ **8.5.5.6.3 Coding lines and curves.** If lines and curves are coded to distinguish among multiple curves on the same graph, the coding shall be used consistently throughout an application and related applications for the same types of data.

□ **8.5.5.6.4 Cyclic data.** If cyclic data are displayed, at least one full cycle should be presented.

□ **8.5.5.6.5 Projected values.** A distinct line code, for example, dashed or dotted lines, should be used to display values projected beyond the actual data set.

8.5.5.7 Areas

□ **8.5.5.7.1 Area between curves.** If emphasis is on the area between two curves, that area should be filled with color or a pattern.

□ **8.5.5.7.2 Stacked curves.** If cumulative data are represented by stacked curves, the curves should be ordered with the least variable at the bottom and the most variable at the top.

□ **8.5.5.7.3 Labeling areas.** Areas in graphs should be labeled within the areas to the extent possible.

8.5.5.8 Scatterplots

□ **8.5.5.8.1 When to use.** Scatterplots should be used to show how individual points are related and distributed along two dimensions.

□ **8.5.5.8.2 Highlighting points.** If a scatterplot contains points of particular importance, those points should be highlighted.

8.5.5.9 Pie charts

□ **8.5.5.9.1 When to use.** Pie charts should be used to show the proportional distribution of categories with respect to the sum of the categories.

□ **8.5.5.9.2 When not to use.** If accurate judgments of magnitudes are required, bar charts should be used rather than pie charts.

□ **8.5.5.9.3 Labeling pie charts.** Pie chart segments should be labeled inside the segments, if possible. Segment labels should include a number stating either the percentage of the whole represented by the segment or the absolute number the segment represents (or both). Labels should be oriented for normal left-to-right reading.

□ **8.5.5.9.4 Highlighting segments.** Segments requiring emphasis should be highlighted or displaced slightly from the rest of the pie chart.

8.5.5.10 Pictures

Pictures are appropriate when a detailed representation of objects is required.

□ **8.5.5.10.1 Automated aids.** If users must perform detailed analyses of images, the application should provide automated aids (for example, the capability to zoom in on a portion of the picture).

8.5.5.11 Diagrams

Diagrams are appropriate if users require information about spatial relationships among objects, but not the level of detail provided by pictures.

□ **8.5.5.11.1 Large diagrams.** If a diagram is too large to view all at once, it should be presented in separate sections, with an overview that indicates the separate sections. Notation should be consistent throughout the diagram. The application should provide an easy means for users to move among the sections.

□ **8.5.5.11.2 Highlighting portions of diagrams.** If portions of a diagram require special attention, those portions should be highlighted.

□ **8.5.5.11.3 Rotation of diagrams.** If users may need to view a diagram from different perspectives, the application should provide the capability of rotating the diagram. The labels of a rotated diagram should be displayed "right-side up" and be legible from the user's perspective.

8.5.5.12 Flowcharts

Flowcharts are appropriate for showing schematic representations of sequential processes and as aids to solving problems if solutions can be reached by answering a series of questions.

□ **8.5.5.12.1 Flowchart design.** Flowchart design should follow one of the following principles:

a.   logical or sequential order, or

b.   minimum path length.

■ **8.5.5.12.2 Consistency.** Words and phrases used for the same purpose shall be consistent throughout a flowchart, an application, and related applications.

□ **8.5.5.12.3 Highlighting.** Paths or portions of a flowchart that deserve particular attention should be highlighted.

□ **8.5.5.12.4 Flowcharts as decision aids.** Flowcharts used as decision aids should require only one decision at each step, and should provide users with a logically ordered list of available options at each step.

□ **8.5.5.12.5 Flowchart orientation.** If possible, flowcharts should be oriented so that paths conform to the following conventions:

a.   left-to-right,

b.   top-to-bottom, or

c.   clockwise.

8.5.6 Display control

8.5.6.1 General

□ **8.5.6.1.1 User tailoring.** Users should be able to tailor information displays by controlling data selection, coverage, updating, and suppression.

□ **8.5.6.1.2 Return to normal display.** If user tailoring of displays is allowed, an easy means should be provided to restore the display to its default displays.

8.5.6.2 Display of control options

■ **8.5.6.2.1 Control locations and options.** Screen control locations and control options shall be clearly and appropriately indicated.

■ **8.5.6.2.2 Default values.** If the system prompts a user for a parameter that has a default value assigned, the default value shall be displayed.

■ **8.5.6.2.3 Control information.** When a control for manipulating the display becomes available, information the user needs for its use shall also be displayed.

8.5.6.3 Data access

□ **8.5.6.3.1 Moving through data.** A consistent and easy means should be provided for moving through a data set, for example, scrolling or paging.

> **Definitions. Scrolling** is a method used to move through the contents of a window or list in a dialogue box using the scroll bar or scroll arrows. **Paging** is process of scrolling through data one page at a time.

□ **8.5.6.3.2 Moving through continuous text.** Scrolling and paging should be provided for moving through continuous text. Panning should not be used.

> **Definition. Panning** is an orientation of display framing in which a user conceives of the display frame as moving over a fixed array of data. The opposite of scrolling.

□ **8.5.6.3.3 Moving through grouped information.** Panning and scrolling should not be used to move through logically grouped information, such as a form.

8.5.6.4 Panning and zooming

□ **8.5.6.4.1 When to provide scrolling, paging, and panning.** If information to be displayed exceeds the available display area, the system should provide a scrolling, paging, or panning capability (see paragraphs 8.5.6.3.2 and 8.5.6.3.3).

□ **8.5.6.4.2 When to provide zooming.** If a user will need to view objects such as pictures, diagrams, or maps in detail, the system should provide a zooming capability.

> **Discussion.** When a portion of a display has been expanded by zooming, it is also desirable to display the portion in its original size and as much of its surrounding context as will fit. Alternatively, the original display might be reduced and displayed with the enlarged portion highlighted.

□ **8.5.6.4.3 Scale indication.** When a portion of a display has been expanded by zooming, the system should provide a scale indicating the amount of expansion.

□ **8.5.6.4.4 Scale integration.** Panning and zooming functions should be integrated with and include scales and other overlaid data, such as scale marks and range vectors.

8.5.6.5 Information suppression

■ **8.5.6.5.1 Suppression indication.** If the display of information is temporarily suppressed, an indication of this suppression shall be provided on the display.

□   **8.5.6.5.2 Indication of changes in suppressed information.** The user should be notified of any significant changes in suppressed information.

■   **8.5.6.5.3 Restoration of suppressed information.** The system shall provide a quick and easy means for restoring suppressed information.

8.5.6.6 Labeling and marking information

■   **8.5.6.6.1 Display identification.** If a system allows users to select and manipulate displays, each display shall have an identifying label and other identifying information to support display control and data access.

■   **8.5.6.6.2 Labels.** Labels that identify displays shall be unique, brief, and meaningful, and they shall be located prominently and consistently.

■   **8.5.6.6.3 Numbering multipage displays.** If information is divided into separate pages for display, each display shall be labeled with the number of the current page and the total number of pages, for example, **Page 3 of 24**.

■   **8.5.6.6.4 Numbering items in multidisplay lists.** If the items in a numbered list do not all fit on one display, the entire set of items shall be numbered continuously; numbering shall not start anew with each display.

8.5.7 Display regeneration and updating

8.5.7.1 General

■   **8.5.7.1.1 Update rate.** If a task requires that a user read changing data, for example, the speed or bearing of an aircraft, individual data items shall be displayed long enough for the user to read them reliably and accurately.

■   **8.5.7.1.2 "Real time" data.** To be considered "real time," changing data that are used for indications of gross values or rate of change shall be updated between two and five times a second.

□   **8.5.7.1.3 Alphanumeric data.** Alphanumeric data that users are required to read reliably and accurately shall not be updated more often than once a second.

8.5.7.2 User control

□   **8.5.7.2.1 Display regeneration.** Unless constrained by task, application, or system requirements, users should be able to initiate display regeneration.

❑ **8.5.7.2.2 User control of rate of update.** Unless constrained by task requirements, users should be able to control the rate of information update.

❑ **8.5.7.2.3 Automatic updating.** If displayed textual data are changed automatically, changed data should be highlighted temporarily or otherwise marked.

8.5.7.3 Freeze frame

❑ **8.5.7.3.1 "Freezing" changing data.** Applications in which displayed data are changed automatically should allow users to "freeze" the display temporarily.

■ **8.5.7.3.2 Labeling a frozen display.** If a display is "frozen," its "frozen" status shall be clearly indicated.

❑ **8.5.7.3.3 Notification of changes while display is frozen.** Users should be notified of any significant changes that occur while a display is frozen.

■ **8.5.7.3.4 Unfreezing a display.** Unless specified otherwise by the user, when a frozen display is released from its frozen state, it shall indicate conditions at the time of release, not the time when it was frozen.

8.5.8 Maps and situation displays

This section contains criteria and guidelines for map and situation displays in general. Criteria and guidelines for map windows are given in section 8.3.12.4.

> **Definitions.** A **map** is a representation of geographic data; a **situation display** is a means of relating dynamic information to a map.

8.5.8.1 General

❑ **8.5.8.1.1 User expectations.** The design of maps and the symbols used in them should be consistent with users' expectations.

❑ **8.5.8.1.2 Amount of detail.** The amount of detail displayed should be consistent with users' operational needs; that is, neither too much nor too little detail should be displayed.

❑ **8.5.8.1.3 Map manipulation tools.** The system should provide users with all appropriate tools for moving easily around a map, including zooming and panning. It should also provide insets, registration, and keys for scale.

❑ **8.5.8.1.4 Curvature.** If large geographic areas are displayed, the curvature of the earth should be treated consistently in all displays.

❑ **8.5.8.1.5 Situation displays as overlays.** Situation displays should be provided as overlays to their related maps.

❏ **8.5.8.1.6 Labeling features.** To the extent possible without cluttering the display, all significant features should be labeled.

❏ **8.5.8.1.7 Consistent label position.** Map labels should be positioned consistently, for example, consistently beneath a feature or consistently within a feature.

❏ **8.5.8.1.8 Consistent orientation.** If more than one map will be displayed, all maps should have the same orientation, for example, with north at the top.

❏ **8.5.8.1.9 Coding areas.** Map areas of special interest should be coded by color or shading. If users must make relative comparisons among areas, shades of a single color should be used rather than different colors. If shades of a color are used, the gradation from light to dark should correspond to the variation represented by the shades.

❏ **8.5.8.1.10 Automated tools.** If users must perform complex analyses of maps, the system should provide the specific, automated tools they need. For example, the system might provide an automated program that prioritizes all alarms displayed on a map.

8.5.8.2 Static display attributes

■ **8.5.8.2.1 Map coverage.** Maps shall cover the areas and display all the essential details users need to perform their tasks. Map displays shall be large enough to permit the simultaneous presentation and visual integration required by users.

■ **8.5.8.2.2 Necessary features.** All features necessary to the completion of the task shall be represented.

■ **8.5.8.2.3 Label legibility.** Labels shall remain legible at all display resolutions.

❏ **8.5.8.2.4 Reducing clutter.** Users should be provided a means for reducing clutter without losing essential information.

❏ **8.5.8.2.5 Association of symbols with map features.** A symbol should be placed accurately with respect to the map feature with which it is associated, or connected to the feature with an arrow, line, or other pointing device so that the association between feature and symbol is clear.

❏ **8.5.8.2.6 Automatic registration.** The system should provide automatic registration of graphic data with background map information at all display scales.

❏ **8.5.8.2.7 Symbol identification key.** Users should have a means for identifying unknown symbols and other map information. For example, a user might be able to highlight a symbol and learn its meaning through a context-sensitive help feature. ·

■ **8.5.8.2.8 Color coding symbols.** Color coding of symbols shall conform to the criteria and guidelines in section 8.5.4.5.

□ **8.5.8.2.9 Nonoverlapping of symbols.** Map symbols should not overlap, particularly if overlapping would obscure their identity. If overlap is unavoidable, users should have a means of moving background symbols to the foreground or otherwise revealing obscured symbols.

□ **8.5.8.2.10 Labeling symbols.** Critical symbols should be labeled automatically. Users should have a means for displaying identifying information about other symbols.

□ **8.5.8.2.11 Coordinate readings.** If location information will be needed frequently, users should have the option of a constant display of the cursor coordinates in units of their choosing. They should also be able to specify coordinates for the placement of an overlay.

□ **8.5.8.2.12 Determining coordinates.** Users should be able to obtain the exact map coordinates of any symbol or map feature.

□ **8.5.8.2.13 Context for displayed map.** If a displayed map is not the entire map, an inset should be provided that shows the entire map with the displayed portion highlighted.

□ **8.5.8.2.14 Determining distances.** Users should be provided with an automated means for determining the distance between two points on a map.

□ **8.5.8.2.15 Determining bearings.** Users should be provided with a means for easily determining the bearing between two points.

8.5.8.3 Dynamic display attributes

□ **8.5.8.3.1 Panning.** If it is required by their tasks, users should be able to move (pan) the viewpoint or window over the entire map in any direction. As long as it meets users' needs, panning may be either continuous or discrete.

□ **8.5.8.3.2 Location information.** Users should be provided feedback during panning operations. For example, the currently displayed portion might be highlighted on an inset display of the entire map.

□ **8.5.8.3.3 Return to start.** If panning is provided, users should have the ability to return to the starting configuration quickly and easily.

□ **8.5.8.3.4 Zooming.** Users should be able to zoom a display in and out, that is, they should be able to increase and decrease the portion of the entire map displayed on the screen.

■   **8.5.8.3.5 Zooming and legibility.** Zooming in and out shall not
    interfere with the ability of users to read symbols, labels, and
    other map features.

    > **Discussion.** It may be appropriate to vary the amount of
    > detail displayed in accordance with the degree of zooming
    > used.

□   **8.5.8.3.6 Discrete vs. continuous zooming.** The method of
    zooming provided, discrete or continuous, should be acceptable
    to the users.

□   **8.5.8.3.7 Return to default.** If zooming is provided, an easy
    means to return to the default display should also be provided.

□   **8.5.8.3.8 Indication of changing scale.** Displays that change
    scale during zooming should include an indicator that shows the
    current scale.

□   **8.5.8.3.9 Indication of displayed portion of map.** A map that is
    capable of being zoomed should include an inset that shows the
    entire map with the currently displayed portion highlighted.

□   **8.5.8.3.10 Selecting information for updating.** If appropriate,
    users should be able to select categories of information that will
    be updated automatically on a map display.

□   **8.5.8.3.11 Stable reference elements.** If a map is updated
    automatically, it should contain some elements that remain stable
    that users can use as reference points.

□   **8.5.8.3.12 Identification of updates.** Users should have a means
    for easily identifying updates and changes to a displayed map. In
    addition, critical changes should be easily distinguishable from
    other changes. For example, critical changes might be highlighted
    and remain highlighted until acknowledged by a user.

□   **8.5.8.3.13 Control of frequency of updating.** Users should be
    able to control the frequency with which a display is updated.

□   **8.5.8.3.14 Rate of updating.** The rate at which a display is
    updated should not exceed the perceptual abilities of its users.

□   **8.5.8.3.15 Freezing a dynamic display.** Users should be able to
    freeze a dynamic display, preventing further updates until the
    display is unfrozen. Frozen displays should include an indication of
    their frozen state. Users should be able to choose to resume
    updating from the time the display was frozen or from the current
    time.

□   **8.5.8.3.16 Control of rate of sequencing.** If appropriate, users
    should be able to control the rate of display sequencing.

    > **Definition. Display sequencing** is a means of reducing
    > clutter by displaying a series of partial displays (for
    > example, a map and a series of overlays) or of displaying

data sequentially. It can also be used as a form of animation.

❑ **8.5.8.3.17 Freezing a sequence.** Users should be able to freeze a sequence, preventing further changes until the sequence is unfrozen. Frozen sequences should include an indication of their frozen state. Users should be able to choose to resume a sequence from the time it was frozen or from the current time.

❑ **8.5.8.3.18 Direction of sequencing.** If appropriate, users should be able to view sequential displays backwards as well as forwards.

❑ **8.5.8.3.19 Viewing selected displays.** Users should be able to return quickly to a selected display in a sequence of displays.

❑ **8.5.8.3.20 Grid overlay.** If appropriate, users should be able to display and remove a grid overlay on a map. If present, a grid should be integrated with the map's coordinate system.

❑ **8.5.8.3.21 Map legend.** Map displays should have associated legends. If the map is dynamic, the legend should change as the map does so that the information is continuously relevant to the current display. This information should include such data as the map scale, cursor location, and status.

8.5.8.4 Creating and editing map graphics

❑ **8.5.8.4.1. Standard symbol library.** Users should have available a library of standard symbols and a means of transferring and manipulating them.

❑ **8.5.8.4.2 Labeling symbols.** Users should have an easy means for labeling symbols.

> **Discussion.** It might be desirable to provide an automated feature that would aid the user in labeling symbols and enforce labeling conventions.

❑ **8.5.8.4.3 Tools for constructing symbols and overlays.** If appropriate, users should be provided tools that would aid them in constructing new symbols and graphic overlays.

❑ **8.5.8.4.4 Editing displays.** If appropriate, users should be able to add to and delete from displays symbols, labels, and other features without destroying background information.

❑ **8.5.8.4.5 Expanding displays.** Users should be able to expand an area of a display if necessary for the accurate placement of critical data.

❑ **8.5.8.4.6 Editing display elements.** Users should be able to perform the following editing operations on elements in map displays:

a.   **Select** elements on the display. Selected elements should be highlighted.

b.   **Move** selected elements on the display.

c.   **Remove** and **Restore** selected elements on the display.

d.   **Name, Store,** and **Retrieve** graphic displays and elements.

□   **8.5.8.4.7 Identifying attributes.** If appropriate, users should be able to identify the currently-selected attributes easily.

□   **8.5.8.4.8 Changing display attributes.** Users should be able to change the attributes of selected display elements.

□   **8.5.8.4.9 Changing display attributes by selection.** Users should be able to change display attributes such as color, symbols, and line types by selecting the attributes from displays rather than by naming the options.

□   **8.5.8.4.10 Print preview.** Users should be able to preview symbols and overlays before printing them.

8.5.8.5 Map display characteristics

□   **8.5.8.5.1 Map visibility.** If important for task performance and to the extent possible, other displays, such as dialog boxes and windows, should not obscure a map display.

□   **8.5.8.5.2 Map cursor.** The cursor in a map display should be a cross hair design that has a high contrast with the background. The cursor should subtend a visual angle of 20 minutes.

□   **8.5.8.5.3 Filters.** Users should be able to reduce the clutter of a map display by "filtering" out such things as overlays, roads, cities, vegetation, and topography. The labels and titles of filters should communicate their function clearly to users.

□   **8.5.8.5.4 Text and overlays.** Text on maps should be integrated with overlays so that the overlay does not obscure the text. If the text is offset from the feature to which it refers, it should be connected to the feature with a line or arrow.

■   **8.5.8.5.5 Color in overlays.** If color is used in overlays, it shall conform to the criteria and guidelines in paragraphs 8.5.4.5.1, 8.5.4.5.2, 8.5.4.5.4, and 8.5.4.5.5.

□   **8.5.8.5.6 Intensity.** The intensity of the map should be controllable to allow the map to be dimmed without losing all the map features.

□   **8.5.8.5.7 Map as background.** If an application uses one map intensively, it is recommended that the map be used as the background or base screen, which should be the maximum display size possible to promote readability.

8.5.9 Voice displays   Voice displays are appropriate (1) to supplement visual displays when communication flexibility is necessary, (2) when coded signal meanings are numerous or may be forgotten, (3) for presentation of complex directions or instructions, (4) when ambient noise may mask simple tonal signals, (5) in conjunction with tonal signals, and (6) for presentation of continuous information when the rate of change is low.

8.5.9.1 Word selection

■ **8.5.9.1.1 Word choice.** The words used in voice displays shall be concise, intelligible, and appropriate to the task and the information presented.

□ **8.5.9.1.2 Words to avoid.** To the extent possible, words that have other words that rhyme with them or that sound similar in other ways should be avoided if these other words might be used in the same context and therefore possibly be confused with the original words.

□ **8.5.9.1.3 "Formal" words.** "Formal" or "correct" words should be used; slang, jargon, and colloquial words should be avoided.

□ **8.5.9.1.4 Alphabetic information.** Alphabetic information should be presented using a phonetic alphabet; that is, words like "alpha," "bravo," and "charlie" should be used rather than the letters "A," "B," and "C."

8.5.9.2 Presentation

□ **8.5.9.2.1 "Average talker."** Spoken messages should sound like an "average talker," that is, one having an American English accent without a regional dialect.

□ **8.5.9.2.2 Distinctive voices.** If different categories of voice signals are used, a different, distinctive voice should be used for each category. For example, one voice might be used for instructional messages and another for warnings.

□ **8.5.9.2.3 Content.** Spoken messages should be brief, informative, and to the point.

□ **8.5.9.2.4 Speech quality.** Speech intensity should be appropriate to the expected ambient noise environment (see section 13.5 criteria and guidelines regarding ambient noise levels). Signal to noise ratio should be at least 5:1. Audio signal power should be approximately 300 milliwatts at the listener's ear.

□ **8.5.9.2.5 Alerting signals.** Spoken warning signals should be preceded by an alerting signal.

□ **8.5.9.2.6 Acknowledging warning signals.** The system should require that users acknowledge spoken warning signals.

8.6 User guidance

This section contains criteria and guidelines for user guidance, including status information, system-initiated routine and error messages, and on-line help. Different types of users may have different needs for user guidance.

a.   **Novices** (users who have little experience with computers) may need help with basic concepts and operations. Novices may want to see only necessary information.

b.   **Experts** (experienced computer users) may want to know about limitations, shortcuts, complex operations, and anything else that will allow them to do their work more efficiently.

c.   **Casual users** may be either novices or experts, but they use the system infrequently and may need to be reminded of aspects of the system they have forgotten.

8.6.1 On-line help

On-line help can provide: (1) procedural aids, (2) the ability to recover from errors, and (3) advice, without requiring a user to exit from the application. Ideally, on-line help is always available and sensitive to the context within which it is requested.

> **Definition. On-line help** is primarily an interactive, context-sensitive source of information that can tell a user what entry to make at the current location in an application, what keystrokes are required, or what steps are required to perform to complete a task. Secondarily, on-line help is a form of on-line documentation and reference information.

An on-line help facility may provide any or all of three types of help: advice, active help, and passive help. Advice is an interactive, context-sensitive source of information that indicates what entry to make at the current location in the application, the required keystroke(s), or which steps to take to complete the task.

Active help senses an inappropriate entry and interrupts the task to ask users what they are attempting and if they are sure they want to complete the operation they have just initiated. Depending upon the user response to the question, active help then suggests the correct action.

Passive help simply responds to user requests for information. The information may be in the form of on-line system documentation, such as a user's guide or a list of functions performed by combinations of keypresses.

8.6.1.1 General

■ **8.6.1.1.1 Applicable criteria and guidelines.** On-line guidance information shall conform to the criteria and guidelines for data display in section 8.5.

□ **8.6.1.1.2 Availability of on-line help.** Specific user guidance information should be available on-line for display at any point in a transaction sequence.

□ **8.6.1.1.3 On-line guidance.** The system should provide users appropriate on-line data, command indexes, and dictionaries to guide them in the selection and composition of data and command entries. This on-line guidance material should include all applicable definitions, lists of allowable entries, ranges of acceptable values, and reference material describing system capabilities and procedures.

□ **8.6.1.1.4 User-centered help.** On-line help should be user-centered, that is, based on the task the user is trying to complete, not on the characteristics of the application.

■ **8.6.1.1.5 Consistent and distinguishable formats.** User guidance shall be displayed consistently in a format that is distinguishable from that of other displayed data.

□ **8.6.1.1.6 Location of displayed help.** To the extent possible, the display of help should not obscure the object about which help was requested. If the help display is in a window, the window should be movable (see paragraphs 8.3.5.3 and 8.3.5.4) and resizable (see paragraphs 8.3.5.6 and 8.3.5.7).

■ **8.6.1.1.7 Highlighting critical information.** Critical information in user guidance shall be highlighted using the same methods used to highlight critical information in other types of data display (see section 8.5.4).

□ **8.6.1.1.8 Prompts.** The system should allow a user to request the display of prompts for the entry of data and command parameters. If supplied, these prompts should be displayed in a standard location, for example, just above the command entry area or the message area. Additional guidance should be available if the simple prompt is not adequate.

□ **8.6.1.1.9 Experienced users.** If the "normal" user guidance techniques provided might slow experienced users, alternative modes should also be provided that allow the bypassing of these "normal" techniques.

□ **8.6.1.1.10 Printing help information.** Users should be able to print displayed help information.

■ **8.6.1.1.11 Searching on-line help.** Users shall be able to search through on-line **Help** displays.

□ **8.6.1.1.12 User annotations.** Users should be able to annotate existing **Help** messages.

□ **8.6.1.1.13 User requests.** Users should be able to request help on selected topics.

8.6.1.2 Access and return

□ **8.6.1.2.1 Access from and return to application.** Users should be able to (1) access help from within an application, that is, without leaving the application, and (2) return to where they were before requesting help.

□ **8.6.1.2.2 Reminder of accessibility.** Users should be reminded constantly of the availability of **Help**. This might be accomplished by the display of the word **Help** in a menu bar or by displaying a push button labeled **Help**.

□ **8.6.1.2.3 Notification of unavailability of help.** If **Help** is not always available, users should be informed when it is not available. This might be done by dimming a **Help** label.

□ **8.6.1.2.4 Standard action.** Users should be able to obtain on-line help by using a standard action that is always available.

□ **8.6.1.2.5 Consistent access.** The procedures for accessing on-line help and, if applicable, for moving from level to level should be consistent throughout an application and related applications.

□ **8.6.1.2.6 Easy access.** Users should not be required to memorize lengthy sequences or refer to secondary written procedures to access on-line help.

■ **8.6.1.2.7 Help command.** The system shall provide a **Help** command that allows users to obtain on-line guidance information.

□ **8.6.1.2.8 Easy alternation between help display and original display.** Users should be able to alternate easily between a help display and the display from which help was requested.

□ **8.6.1.2.9 Easy return.** After requesting and receiving help, a user should be provided with an easy means to return to the display from which help was requested. For example, a user should not have to call up a menu and select an option to return from a help display.

□ **8.6.1.2.10 Control options.** Any help or guidance display should include any relevant control options. For example, a help window might include an **OK** push button for removing the window.

■ **8.6.1.2.11 Single action.** Users shall be able to access and exit **Help** with a single action, for example a single keystroke or a single click of a pointing device.

□ **8.6.1.2.12 Marking topics for retrieval.** If the number of topics in an on-line help facility is large, and if it would be useful to users to be able to customize the facility by marking individual topics for retrieval, the facility should provide this capability. That is, users should be able to mark individual topics and then retrieve only the marked topics.

□ **8.6.1.2.13 Synonyms.** Synonyms for standard terminology should be recognized by help routines.

8.6.1.3 Context sensitivity

■ **8.6.1.3.1 Task-oriented help.** The information provided in response to a **Help** request shall be relevant to the task and the current transaction within the task.

□ **8.6.1.3.2 Ambiguous context.** If the context in which a request for help is made is ambiguous, the system should initiate a dialog in which the user can specify what data, message, or command requires explanation.

□ **8.6.1.3.3 Context information in help display.** If a user's request for help depends upon the context established by previous entries, an indication of that context should be included in the help display.

□ **8.6.1.3.4 List valid entries.** If possible, when a user makes an invalid entry, the system should provide a list of valid entries.

□ **8.6.1.3.5 Historical context.** If appropriate, users should be able to request a displayed record of past transactions.

8.6.1.4 Wording and style

■ **8.6.1.4.1 Applicable criteria and guidelines.** The wording and style of on-line help shall conform to the criteria and guidelines in sections 10.2.3 and 10.2.4.

□ **8.6.1.4.2 Wording.** The wording of help information should be brief, specific, and task-oriented. If appropriate, it may incorporate special terms and technical jargon that are normally employed in the user's tasks.

■ **8.6.1.4.3 Appropriate to user.** Help information shall be appropriate to the experience and training of the system users.

8.6.1.5 Content

□ **8.6.1.5.1 Scope.** On-line help should include (1) memory aids, (2) basic information likely to be of use only to novices, (3) material selected from written documentation, (4) explanations that go beyond written documentation, (5) information that might seem obvious, but may not be to all users, and (6) step-by-step instructions on how to perform the most common tasks.

□   **8.6.1.5.2 Only relevant information.** Help displays should contain only information relevant to the current requirements of the user.

□   **8.6.1.5.3 Multilevel help.** The system should all provide multiple levels of help, with successive levels providing increasingly detailed levels of explanation.

□   **8.6.1.5.4 Help on Help.** On-line help should include help on how to use the on-line help. This help should include:

   a.   a description of all **Help** displays,

   b.   instructions on how to access **Help** from anywhere in the system, including alternative routes, if any,

   c.   instructions on navigating through **Help**, including scrolling, paging, and moving to related topics, if applicable, and

   d.   a description of the current window, including its function and any tasks the user can perform.

■   **8.6.1.5.5 Titles.** Each **Help** display shall have a title that identifies its contents and reflects the location from which it originated.

□   **8.6.1.5.6 System information.** On-line help should include a description of system capabilities and procedures.

□   **8.6.1.5.7 Application information.** On-line help should include a description of the application, including its capabilities, components, options, and structure.

□   **8.6.1.5.8 Available commands.** If an application uses commands, an on-line index and description of all commands should be available.

□   **8.6.1.5.9 Command examples.** If appropriate, help displays should include examples of correct input or valid commands. Examples should include realistic commands and parameters, not just formal syntax.

□   **8.6.1.5.10 Command format.** If appropriate, help displays should include a description of the format of a specified command and a list of allowable commands.

□   **8.6.1.5.11 Function keys.** On-line help should provide multilevel descriptions of the actions assigned to function keys.

□   **8.6.1.5.12 Prompts, requests, and definitions.** On-line help should provide multilevel help on any displayed prompts or requests and definitions of all important terms.

□   **8.6.1.5.13 Error messages.** On-line help should provide multilevel help on error messages.

□ **8.6.1.5.14 Shortcuts.** On-line help should point out shortcuts and infrequently used features to users.

□ **8.6.1.5.15 Help index.** An on-line index of help topics should be available to users.

□ **8.6.1.5.16 Finding Help topics.** The on-line help facility should allow users to press any alphabetic key and obtain a list of the help topics beginning with that letter. Users should then be able to select a topic from the list and obtain the help information for that topic.

8.7 Data communication

This section is concerned with communication among users within a single computer system and among users on different interconnected systems. While data communication includes the transmission of all sorts of data among users, for example, text files and graphic files, this section is concerned primarily with the exchange of formatted messages. Special considerations and restrictions that apply to "sensitive" or classified information are given in section 11. Additional information is available in section 8.6.9.2, Message windows.

8.7.1 User control and procedures

■ **8.7.1.1 Integration with other system functions.** Data transmission functions shall be integrated with other information handling functions within a system.

■ **8.7.1.2 Consistent procedures.** Procedures for preparing, sending, and receiving messages shall be consistent between transactions and other information handling tasks.

■ **8.7.1.3 Minimal memory load.** Data transmission procedures shall be designed to minimize memory load on the user and to minimize required user actions.

■ **8.7.1.4 Explicit user actions.** Both sending and receiving messages shall be accomplished by explicit user action.

□ **8.7.1.5 User control.** Users should be in control of what, when, and where data are transmitted.

□ **8.7.1.6 Interruptible by user.** Users should be able to interrupt message preparation, review, or disposition. Resumption should be from the point of interruption.

■ **8.7.1.7 Annotations to transmitted data.** Transmitted data shall be annotated with any alarm or alert conditions, priority indicators, and other significant information that exist.

8.7.2 Preparing messages

8.7.2.1 General

■ **8.7.2.1.1 Applicable criteria and guidelines.** The procedures for composing messages shall conform to section 8.3. To the extent possible, the procedures for entering messages should be the same as those for general data entry.

□ **8.7.2.1.2 Printing messages.** Users should be able to print copies of transmitted messages.

8.7.2.2 User control

□ **8.7.2.2.1 Length of messages.** Users should be able to prepare and transmit messages of any length.

□ **8.7.2.2.2 What can be transmitted.** Users should be able to specify the data to be transmitted. They should be able to incorporate existing file data (including other messages received or transmitted) into messages (see also paragraph 8.7.4.2.2).

□ **8.7.2.2.3 Saving prepared messages.** Users should be able to save draft messages during preparation and after completion.

8.7.2.3 Message format

□ **8.7.2.3.1 User-designed format.** Unless a need exists for a specific message format, users should be able to compose and transmit messages with a format of their own design, and also to compose and transmit messages as unformatted text.

■ **8.7.2.3.2 Application-supplied format.** If messages must conform to a defined format, a preformatted message form shall be available to users (see section 8.4.2).

8.7.3 Addressing messages

8.7.3.1 User control

□ **8.7.3.1.1 User-specified destinations.** Users should be able to specify destinations to which data will be transmitted. Destinations may include individuals, groups of individuals, work stations, terminals, and remote printers.

□ **8.7.3.1.2 Editing address fields.** Users should be able to edit the address fields in the header of a message being prepared for transmission.

□ **8.7.3.1.3 Listing other users on-line.** The system should provide users the capability of listing the other system users who are currently on-line.

8.7.3.2 Message formatting

□ **8.7.3.2.1 Message header fields.** A basic set of labeled message header fields should be provided, including **Date, To, From, Copy to,** and **Time.** These fields should be interpretable by all systems to which messages can be sent.

□ **8.7.3.2.2 Prompting.** Prompting should be provided to guide the user in specifying the address for a message.

8.7.3.3 Directories and distribution lists

□ **8.7.3.3.1 On-line directories.** On-line address directories should be provided in which users can search for addresses by specifying a complete or partial name, or other address information. Users should be able to select addresses from a directory for automatic entry in address fields.

□ **8.7.3.3.2 Substitute addresses.** Users should be able to define substitute addresses for commonly used addresses and use these substitutes to address messages. For example, a user might define "jane" to stand for the address "jdoe@smtplink.cta.com."

□ **8.7.3.3.3 User-defined distribution lists.** Users should be able to create and modify their own lists of addressees. They should be able to include the names of distribution lists as well as the names of individual addressees.

8.7.3.4 Validation and error correction

□ **8.7.3.4.1 Valid address.** To the extent possible, the system should ensure that an address is valid.

Examples. If an address is internal to a system, the system might search an on-line directory to validate the address. If an address is external, the system might ensure that the address contains a valid gateway or that the address format is valid.

□ **8.7.3.4.2 Error correction.** The system should prompt users to correct any errors it detects before initiating message transmission.

8.7.4 Initiating transmission

8.7.4.1 System control

□ **8.7.4.1.1 Automatic queuing of outgoing messages.** If a system cannot transmit an outgoing message immediately, it should place

the message in a queue and automatically make repeated attempts to transmit it. The user should not have to makerepeated attempts.

□ **8.7.4.1.2 Appended information.** When a message is sent, the computer should automatically append the sender's address, and the date and time of message creation and transmission.

8.7.4.2 User control

□ **8.7.4.2.1 User initiation of data transmission.** Data transmission should be initiated by an explicit user action, for example, a Send command (see also paragraph 8.7.1.4).

□ **8.7.4.2.2 What users can transmit.** Users should be able to transmit both information that is displayed on their screens and information stored in files (see also paragraph 8.7.2.2.2).

□ **8.7.4.2.3 User-assignable priority.** If a system is capable of treating messages differently based on priority, users should be able to assign a priority to messages.

□ **8.7.4.2.4 User-specified delivery.** In addition to immediate transmission, users should be able to specify other times when a message will be transmitted. This specification might include date, time, or upon the occurrence of a specified event.

□ **8.7.4.2.5 Notification of transmission and delivery.** Users should be able to request notification that a message has been transmitted and that it has been opened by the addressee.

□ **8.7.4.2.6 Cancellation of undelivered messages.** Users should be able to cancel a message that has not been completed and a message that has not been transmitted.

8.7.5 Controlling transmission

8.7.5.1 System control

□ **8.7.5.1.1 Transmitted message log.** If required, the system should automatically maintain a record of transmitted messages.

8.7.5.2 User control

□ **8.7.5.2.1 User-specified feedback.** Users should be able to specify what feedback will be provided for message transmission, and to request specific feedback for particular messages (see also paragraph 8.7.4.2.5).

□ **8.7.5.2.2 Cancellation of messages after initiation.** Users should be able to cancel a transmission after initiation, if the message has not been received.

8.7.5.3 Transmission failure

□ **8.7.5.3.1 Automatic queuing.** If a transmission attempt fails, the system should automatically queue an outgoing message and make subsequent attempts to transmit the message.

□ **8.7.5.3.2 Transmission failure.** If message transmission fails, automatic storage of undelivered messages should be provided, and the sender should be notified. Notification should, if possible, include an explanation of the failure.

8.7.6 Receiving messages

8.7.6.1 System control

□ **8.7.6.1.1 Incoming message queuing.** Incoming messages should be queued automatically by time of receipt, message priority, or user specification, pending subsequent review and disposition by the user.

□ **8.7.6.1.2 Incoming message log.** The system should automatically maintain a log of incoming messages.

8.7.6.2 User control

□ **8.7.6.2.1 User control of incoming messages.** Users should be able to specify data that may be received by specifying receipt priority or other characteristics, and they should be able to choose the device (files, display, printer) that will receive messages.

□ **8.7.6.2.2 User control of notification of incoming messages.** Users should be able to specify "filters" based on message source, priority, type, or content, that will control their notification of incoming messages.

□ **8.7.6.2.3 Naming and describing incoming messages.** Users should be able to assign their own names and other descriptors to received messages.

□ **8.7.6.2.4 Disposing of incoming messages.** Users should be able to discard unwanted messages without saving them.

8.7.6.3 User review of messages

□ **8.7.6.3.1 User specification of summary order.** Users should be able to specify the order in which message summaries are listed.

□ **8.7.6.3.2 User review of summary information.** Users should be able to review summary information (for example, the source, type, and priority) about queued incoming messages.

▫ **8.7.6.3.3 Nondestructive review.** Unless precluded by security or
other considerations, users should be able to review messages in
their incoming queues without having to dispose (for example, save,
delete, or respond) of them.

■ **8.7.6.3.4 Applicable criteria and guidelines.** The way in which
incoming messages are displayed shall conform to the criteria and
guidelines in section 8.4.4.

▫ **8.7.6.3.5 Annotating incoming messages.** Users should be able to
annotate reviewed messages. Annotations should be displayed and
should be distinct from the message itself.

▫ **8.7.6.3.6 Size indication.** The message summary should include an
indication of the size of the message. This indication should also
be included at the beginning of a message.

8.7.6.4 Incompatible
data format

■ **8.7.6.4.1 Data preservation.** The arrival of a message in a format
incompatible with that of the system shall not result in the loss of
the message or of any ongoing transaction.

▫ **8.7.6.4.2 Notification of incompatible format.** If the format of a
data transmission is incompatible with the system receiving it
(for example, incompatible with system decoding or with the
available devices), the intended recipient should be notified.

8.7.6.5 Notification of
incoming messages

▫ **8.7.6.5.1 Notification at log on.** Users should be notified at log on
of any data transmissions received since their last use of the
system.

■ **8.7.6.5.2 Noninterfering notification.** Notifying a user of an
arriving message shall not interfere with any ongoing transaction.

▫ **8.7.6.5.3 Priority of incoming messages.** If incoming messages
will have different degrees of urgency, the messaging system
should provide users the ability to assign a priority to a message,
and the priority assignment should be indicated for the incoming
message.

8.7.6.6 Replying to a
message

▫ **8.7.6.6.1 Automatic addressing of replies.** If a user replies to a
message, the messaging system should provide the appropriate
address(s) automatically.

8.8 Input devices

This section provides criteria and guidelines for keyboards, function keys, pointing devices, and some alternative input devices.

The advantages and disadvantages of non-keyboard input devices are shown in exhibit 8.8. The characteristics of these devices need to be considered in the selection of the appropriate controls for a given task.

Exhibit 8.8 Advantages and disadvantages of non-keyboard input devices

Type of controller	Advantages	Disadvantage
Mouse	Relatively fast	Requires additional flat work surface
	Has low error rates for large targets	Difficult to use for free-hand graphic input
		High error rates with small targets
	Allows user to concentrate attention on VDT screen	Lost time when mouse held backwards or sideways
		Some training needed
		Wheel (ball) slipping sometimes a problem
Directional controllers (joystick and trackball)	Can be used comfortably with minimum arm fatigue	Slower than a light pen and other "point-to devices" for simple input and option selection.
	Does not cover parts of the screen in use	Must be attached, but not to the display.
	Expansion or contraction of cursor movement is possible	Unless there is a large joystick, an inadequate control to display ratio will result for positional control.
	Ball control is an efficient use of space	The displacement of the stick controls both the direction and the speed of cursor movement.
		Trackball and joystick controllers are difficult to use for accurate free-hand graphic input.
		Difficult to integrate the activate switch with the trackball.

Exhibit 8.8 (continued) Advantages and disadvantages of non-keyboard input devices

Type of controller	Advantages	Disadvantage
Light pen	Fast for simple input	May not feel natural to user, like a real pen or pencil does
	Good for tracking moving objects	Requires some fine motor control
	Minimal perceptual motor skills needed	May lack precision because of the aperture, distance from the CRT screen surface, and parallax
	Efficient for successful multiple selection	Contact with the computer may be lost unintentionally
	User does not have to scan to find a cursor somewhere on the screen	Frequently required simultaneous button depression may cause slippage and inaccuracy
	May be adaptable to bar coding	Must be attached to terminal, which may be inconvenient
		Glare problem if pen tilted to reduce arm fatigue
		Fatiguing if pen is held perpendicular to work surface
		If pointed to dark area, may require user to flash the screen to find pen
		One-to-one input only (zero order control)
		May be cumbersome to use with alternate, incompatible entry methods, like the keyboard
		Tends to be used for purposes other than originally intended, such as for key depression
		Tends to be fragile
		Hand may obstruct a portion of screen when in use
		Care must be taken to provide adequate "activate" area around choice point
		Cannot be used on gas plasma panel
Stylus and grid	Good for graphic entry	Extra space required on work surface
	Can be designed to be used on horizontal surface	Displacement of visual feedback from motor activity may cause coordination problems
	Multipurpose input device	Entering handprinted characters to be recognized by the system is very slow (fewer than 40 characters/min) compared with typewriter entry (averaging 200 recognition characters/min)
	Minimal difficulty going from graphic input if character is built into the system, and the tablet is used for the input	
	Spatial correspondence between displays and control movement	

Exhibit 8.8 (continued) Advantages and disadvantages of non-keyboard input devices

Type of controller	Advantages	Disadvantages
Touch screen	No separate input device needed	Low resolution
	Fast	Finger can block view
		Fingerprints on screen
		Tires arm
Voice activation	Does not require hands	Entry can be slow
	Does not require user to shift gaze	Must use specified vocabulary
	Useful for low light conditions	Some systems must be individualized to specific user
	Allows simultaneous activation of more than one control mode	If individual's voice changes (for example, become stressed) system may not respond
		May require headset
	Could be used in lieu of a translator, allowing natural, conversational version of different languages to control complicated systems	Speaker-dependent systems require template loading time

8.8.1 Keyboards

Keyboards vary greatly in the number and arrangement of keys. Most keyboards include the following:

Note. In this section, when the name of a key that appears on a keyboard is used, it is printed in **Univers Bold** type.

a. Alphanumeric keys. The letters of the alphabet, numerals, and punctuation symbols (numeric keypads may be separate on portable computers).

b. Dedicated formatting keys. Keys for text formatting operations such as a **Space bar,** a **Tab** key, and a **Return** or **Enter** key.

c. Modifier keys. Keys that modify or qualify the effects of other keys for as long as they are held down, for example, **Shift, Ctrl,** and **Alt.**

d. Navigation keys. Keys that move a cursor, for example, arrow keys, **Home, End, Page Up,** and **Page Down.**

e. Fixed-function keys. Keys provided for extra or general functions, typically labeled **F1, F2,** and so on.

f.   Special purpose keys. Keys that have a special function,
        such as **Help, Delete,** and **Backspace.**

■ **8.8.1.1 When to use.** If applicable, keyboards shall be provided for
the entry of alphabetic, numeric and other special characters into
the system. Keyboards shall conform to ANSI/HFS 100-1988,
unless otherwise specified or approved by the acquisition
program office.

■ **8.8.1.2 Numeric keypads.** If an application requires substantial
and repetitive input of numeric data, the keyboard shall include a
numeric keypad.

■ **8.8.1.3 VDT keyboard layout and features.** VDT keyboard
layout and features (such as key shape, spacing, force, and height)
shall conform to ANSI/HFS 100-1988.

□ **8.8.1.4 Standard keyboards.** If feasible, standard keyboards
should be used. Nonstandard keyboards should contain only
those keys that are used by the keyboard user.

> **Discussion.** The presence of nonrelevant keys, such as
> those that might be used by programmers, adds to
> keyboard complexity and may induce errors.

■ **8.8.1.5 Cursor movement keys.** Cursor movement keys shall be
arranged in a spatial configuration reflecting the direction of
actual cursor movement. Exhibit 8.8.1.5 shows the arrangement of
cursor movement keys.

Exhibit 8.8.1.5 Cursor movement keys
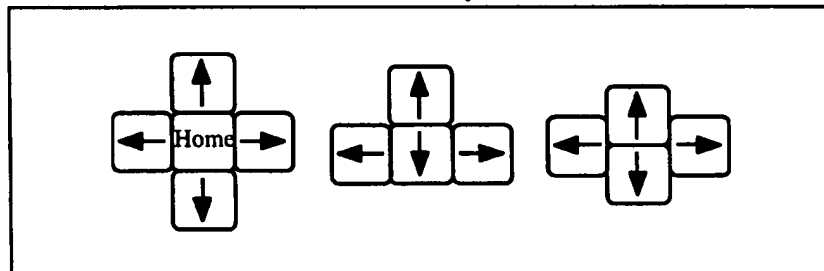
■ **8.8.1.6 Changing data.** Users shall be provided a means to
change previous entries by delete, backspace, and insert actions.

□ **8.8.1.7 Keyboard equivalents to function keys.** If an
application assigns operations to function keys, the operations
that can be performed with a function key should also be
performable with alphanumeric keys.

□ **8.8.1.8 Keyboard equivalents to pointing device operations.** If an
application provides both a keyboard and a pointing device, the
operations that can be performed with the pointing device should
also be performable with the keyboard (see also paragraph 8.8.5.1).

8.8.2 Fixed-function keys

□ **8.8.2.1 Standardization.** Fixed-function keys should be standardized throughout the system.

□ **8.8.2.2 Availability.** Fixed-function keys should be selected to control functions that are continuously available; that is the lock out of fixed-function keys should be minimized. Mechanical overlays should not be used to lock out function keys.

□ **8.8.2.3 Nonactive keys.** If a keyboard is dedicated for use with only a specific application, nonactive fixed-function keys should be replaced by blank keys on the keyboard.

■ **8.8.2.4 Grouping.** Fixed-function keys shall be grouped logically and shall be placed in distinctive locations.

8.8.3 Pointing devices

This section contains criteria and guidelines for pointing devices in general, the shape of the pointer itself, and buttons on pointing devices.

> **Definitions.** A **pointing device** is a non-keyboard device that allows a user to navigate rapidly around the screen and to specify and select objects for manipulation and action. Examples include a mouse, trackball, stylus and grid, and light pen. A **pointer** is a symbol displayed on the screen that is controlled by a pointing device. Its shape may change depending on the function that is invoked at a particular moment or its location on the screen.

8.8.3.1 General

■ **8.8.3.1.1 Functionality.** If present, a pointing device shall be capable of (1) moving a pointer on the screen, (2) selecting objects on which the pointer is placed, and (3) drop and drag operations.

■ **8.8.3.1.2 Single pointer.** A pointing device shall be associated with a single pointer on the screen.

■ **8.8.3.1.3 Moving the pointer.** A user shall be able to move the pointer on the screen by moving all or part of the pointing device. The pointer shall move in the same direction that the pointing device moves. A user shall be able to move the pointer anywhere on the screen.

■ **8.8.3.1.4 Nondisappearance of pointer.** A pointer shall not move beyond the outer boundaries of the screen, nor shall it disappear from sight.

> **Exception.** If there is another screen adjacent to the first, the pointer may move from one screen to the other. This rule does not apply when a cursor is moved quickly and

the screen refresh rate is too slow to show the full path of the cursor.

▫ **8.8.3.1.5 Control of the pointer.** A pointer should not move on the screen unless a user moves the pointing device. That is, an application should not move a pointer arbitrarily.

Exceptions. One exception to this rule is if an application automatically moves the pointer in conjunction with the scroll bar. For example, when the user clicks on the down arrow to scroll through a document, the application may automatically move the pointer so that the pointer will remain on the scroll arrow.

Another case may be when the pointer "jumps" or "snaps-to" a default button because the user has selected that default option.

■ **8.8.3.1.6 Pointer stability.** The stability of the pointer shall be within 1.3 mm (0.05 in) in any direction; the preferred stability is within 0.25 mm (0.01 in).

▫ **8.8.3.1.7 Movement ratio.** The ratio of movement of the pointing device to the movement of the pointer should default to approximately 1:1 and be adjustable by the user.

▫ **8.8.3.1.8 Type of device.** The pointing device selected for an application should be the one most appropriately meets the application requirements and is most cost-effective. The appropriateness of some specific types of pointing devices for tasks is as follows:

a.   A **mouse** is a general purpose pointing device suitable for a wide range of applications.

b.   A **joystick** is appropriate for tasks requiring precise adjustments and continuous control.

c.   A **trackball** is appropriate for generating precise X and Y output values and cumulative travel in any direction.

d.   A **light pen** is appropriate for noncritical, imprecise functions, especially if the primary task is item selection.

e.   A **stylus and grid** is appropriate for graphic entry.

Discussion. Another factor that may contribute to the appropriateness of a given input device is the expectations, experiences, or preferences of the intended user population. If a given user population has a wealth of experience, familiarity, or acquired skill with a particular type of device, careful consideration needs to be given to replicate the features, functionality, performance, and "feel" to which they are accustomed.

8.8.3.2 Mouse

□ **8.8.3.2.1 Use.** A mouse (also known as a free-moving X-Y controller) should be used for zero order control only (for example, the generation of X and Y outputs by the controller results in proportional displacement of the pointer).

> **Discussion.** This type of pointing device may be used on any flat surface to generate X and Y coordinate values that control the position of the pointer on the associated display. It may be used for data pick off or for entry of coordinate values.

■ **8.8.3.2.2 Dynamic characteristics.** The design of the mouse and the placement of the maneuvering surface shall allow the user to consistently orient the mouse within 10° of the correct orientation without visual reference to the mouse.

> **Discussion.** If the user grasps the mouse in what seems to be the correct orientation and moves it rectilinearly along what is assumed to be straight up the Y-axis, then the direction of movement of the cursor on the CRT is to be between 350° and 10°.

■ **8.8.3.2.3 Easily moved.** The mouse shall be easy to move in any direction without a change of hand grasp.

■ **8.8.3.2.4 Lateral range.** A complete lateral movement of the mouse from side to side within the maneuvering area (such as a mouse pad) shall move the pointer from side to side on the display regardless of the scale setting or offset unless expanded movement is selected for an automatic sequencing mode of operation. Users shall be able to specify or modify the lateral movement ratio.

■ **8.8.3.2.5 Dimensions and shape.** The mouse shall have no sharp edges but shall be shaped roughly as a rectangular solid, with limiting dimensions as shown in exhibit 8.8.3.2.5.

Exhibit 8.8.3.2.5 Dimensions of a mouse

Dimension	Minimum mm (in)	Maximum mm (in)
Width (spanned by thumb to finger grasp)	40 (1.6)	70 (2.8)
Length	70 (2.8)	120 (4.7)
Thickness	25 (1.0)	40 (1.6)

8.8.3.3 Joystick and trackball

Joysticks and trackballs are appropriate to use if precise input functions are required. They are most useful when used to

control direct pointing, rather than discrete controls such as cursor control keys.

■ **8.8.3.3.1 Use and conformity.** A joystick and trackball shall conform to sections 7.4.4.17 through 7.4.4.22.

■ **8.8.3.3.2 Activation and deactivation.** A discrete mechanism shall be provided to allow the user to activate and deactivate the joystick or trackball.

8.8.3.4 Light pen

A light pen is appropriate to use if item selection is the primary type of data entry. For example, a light pen may be used when noncritical, imprecise input functions are required. It may also be used as a track-oriented readout device. It can be positioned on the display screen to detect the presence of a computer-generated track by sensing its refresh pattern. The display system will then present a cursor on the designated track. With suitable additional circuitry, a cursor can be made to track the movement of the light pen across the surface, thus allowing it to function as a two-axis controller capable of serving the same purposes as stylus and grid devices (see section 8.8.3.5).

■ **8.8.3.4.1 Dynamic characteristics.** If a light pen is used as a two-axis controller, it shall conform to section 8.8.3.4.

■ **8.8.3.4.2 Dimensions and mounting.** A light pen shall be between 120 and 180 mm (4.7 and 7.1 in) long with a diameter between 7 and 20 mm (0.3 and 0.8 in). A clip shall be provided to hold the light pen when it is not in use.

■ **8.8.3.4.3 Activation.** A light pen shall be equipped with a discrete activating and deactivating mechanism. A push-tip switch, requiring between 0.5 to 1.4 N (2 to 4 oz) of force to activate, is preferred.

■ **8.8.3.4.4 Feedback.** Two forms of feedback shall be provided to the user when using a light pen.

a. Feedback concerning the position of the light pen, preferably in the form of a displayed cursor or highlighting, that informs the user that the system is recognizing the presence of the light pen. The feedback shall be large enough to be seen under the point of the light pen.

b. Feedback that the light pen has been activated (for example, the push-tip switch has been triggered) and the input has been received by the system.

8.8.3.5 Stylus and grid

A stylus and grid is appropriate to use as a multipurpose input device when combined with a program for character recognition. The stylus and grid are also very good for graphic entry although they are much slower than keyboard entry for alphanumeric data.

■ **8.8.3.5.1 Refresh rate.** The refresh rate for the cursor shall be sufficiently high to ensure the appearance of a continuous track whenever the stylus is used to generate free-drawn graphics.

■ **8.8.3.5.2 Remote grid size.** A remote grid shall approximate the size of the display.

■ **8.8.3.5.3 Remote grid placement.** A remote grid shall have an orientation that is consistent with the directional relationships between them and the display without violating any anthropometric criteria and guidelines (see also Section 14 for anthropometric and biomechanical considerations).
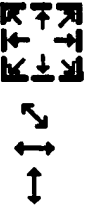
8.8.3.6 Pointer shapes

■ **8.8.3.6.1 General-purpose pointer shape.** An arrow pointing up and to the left shall be the general-purpose pointer (κ). This and other examples of pointer shapes associated with specific functions are illustrated in exhibit 8.8.3.6.1. If an application provides any of these functions, it shall change the pointer to the associated shape whenever that function is invoked. An application shall redefine the shape of a pointer only when the pointer is inside an application window (including the border).

■ **8.8.3.6.2 "Hotspot."** A pointer shall have a "hotspot," that is an active point (although this active point may not be readily apparent to the user). The hotspot shall indicate the precise location where an operation will occur. These points are specified for a variety of pointer shapes in exhibit 8.8.3.6.1.

> **Definition.** A **hotspot** is the precise part of a screen pointer that marks the screen position where an operation on a pointing device will have an effect.

■ **8.8.3.6.3 Hotspot and pointer shape.** The screen location of a hotspot shall not change if the pointer changes from one shape to another.

□ **8.8.3.6.4 Additional pointer shapes.** If an application provides a function for which a pointer shape does not exist in exhibit 8.8.3.6.1, the application may provide a new pointer shape. If this is done, the new shape should (1) be easy to see, (2) obscure as little information as possible on the screen, (3) have a hotspot that is obvious and easy to locate, (4) provide a hint of its purpose, and (5) not be easily confused with other objects on the screen.

Exhibit 8.8.3.6.1 Pointer shapes associated with functions

Shape	Name	Function	Hotspot
▸	Arrow	**Pointing.** Used in most window areas for object selection.	The point of the arrow.
I	I-beam	**Pointing.** Used in text areas to position the text cursor and perform actions on text. The I-beam pointer is hidden during the time between any keyboard action and pointer movement (that is, when text entry is occurring at the location of the text cursor).	On the vertical bar of the I-beam about one-third from the top.
⌀	Watch (or hourglass)	**Working.** Indicates that an operation is being performed in a window area. When the working pointer is displayed, all pointing device and keyboard actions are ignored in the area.	Not applicable
⊖	Caution sign	**Caution.** Indicates that action is expected in another window area before input can be made in the current area and that the pointer has no effect in the area. When the caution pointer is displayed, all pointing device and keyboard actions are ignored in the area.	Not applicable
↖↑↗ ←→ ↙↓↘ ↘ ↔ ↕	Resize pointer	**Resize.** Indicates positions for area resize, with the direction of the arrow in the pointer indicating the direction of increasing size. The horizontal and vertical resize pointers indicate resize in either the horizontal or vertical direction. The diagonal resize pointers indicate resize in both the horizontal and vertical directions simultaneously. The resize pointer appears when the pointer is on the frame border.	On the corner or line at the position pointed to by the arrow.
✛	Move arrows	**Moving.** Indicates a move operation in progress or a resize operation before the resize direction has been determined. During a resize operation, the four-directional arrow pointer indicates a direction for resizing and changes to the appropriate resize arrow when the pointer is on the frame border.	The intersection of the arrows.
+	Sight or cross	**Sighting.** Used to make fine position selections (for example, to select a location on a map display).	The intersection of the lines.

8.8.3.7 Pointing device buttons

One or more buttons are provided on pointing devices to allow the manipulation of objects on the screen.

■ **8.8.3.7.1 Button operations.** A user shall be able to perform the following actions with any button on a pointing device:

a. Press. Depress a button and hold it down.

b. Release. Release a button that has been depressed.

c. Click. Press and release a button without moving the pointing device.

d. Double click. Press and release a button twice in rapid succession without moving the pointing device.

e. Drag. Depress a button and move the device while holding the button down.

f. Move. Move the pointing device without pressing any buttons.

■ **8.8.3.7.2 Button functions.** Each button on a pointing device shall have a specific function (within the context of the application) that is executed whenever a user presses the button. If the device has only one button, that button shall provide the "select" function; if it has two buttons, the left one shall provide the "select" function and the right button shall provide a "menu" function.

> **Definitions.** The **select function** selects or activates objects on the screen or sets the location of the cursor. The **menu function** causes the appearance of a menu appropriate to the location of the pointer.

> **Discussion.** If applicable, a system may require that a middle button be used for a particular function (for example, as another means to execute a default action). An application can map a function to the middle button if the function does not contradict or interfere with the function assigned to this button by the system or by another application.

■ **8.8.3.7.3 Left-right reversal.** A system shall provide users the ability to reverse the left-right operation of the buttons.

8.8.4 Alternative input devices (non-keyboard, non-pointing devices)

Application developers are encouraged to use input devices in unique ways to support efficient user performance within an application. In addition, developers might determine that devices such as voice input or touch panels are appropriate alternatives for user input.

8.8.4.1 General

▫ **8.8.4.1.1 Consistent interaction.** If an alternate input device is used in an application, the manner in which users interact with

the device (for example, for navigation or selection), should be consistent with their interactions with other input devices.

■ **8.8.4.1.2 Type of device.** The alternate input device selected for an application shall be the one that most appropriately meets the application requirements and is most cost-effective. The appropriateness of some specific types of input devices for tasks is as follows:

a. A **touch screen or touch panel** is appropriate for data entry and item selection if typing skills are not required.

b. An **optical character recognition device** is appropriate for the entry of formatted, printed data.

c. A **voice input** device is appropriate if the user's visual and manual performance are constrained.

8.8.4.2 Touch panels

□ **8.8.4.2.1 Use.** A touch panel or screen should be used to provide an overlaying control function to a display device (for example, a CRT, an electroluminescent display, or a programmable indicator) if direct visual reference access and optimum direct control access are desired.

□ **8.8.4.2.2 When not to use.** A touch panel or screen should not be used when high resolution monitors are needed or if the user will be making large amounts of data input.

> **Discussion.** Touch panels and screens have low resolution or diminishes the user's ability to see through a touch membrane. Additionally, users' arms often become tired when they have to use touch panels or screens over an extended period of time due to the lack of arm, hand, or wrist support.

■ **8.8.4.2.3 Luminance transmission.** Touch panels shall have sufficient luminance transmission to allow the display to be clearly readable in the intended environment.

■ **8.8.4.2.4 Positive indication.** A positive indication of touch-panel activation shall be provided to acknowledge the system response to the control action.

■ **8.8.4.2.5 Dimensions and separation.** The dimensions and separation of responsive areas of the touch panel shall not exceed the maximum and minimum values given in exhibit 8.8.4.2.5.

> **Note.** The maximum values listed in the exhibit apply to logically grouped touch panel responsive areas. An adverse environment may warrant larger sizes and separations.

■ **8.8.4.2.6 Resistance.** The force required to operate force-activated touch panels shall conform to ANSI/HFS 100-1988.

Exhibit 8.8.4.2.5 Touch panel responsive area dimensions

	A Size	B Separation
Minimum	19 mm (0.75 in)	3 mm (0.13 in)
Maximum	38 mm (1.5 in)	6 mm (0.25 in)
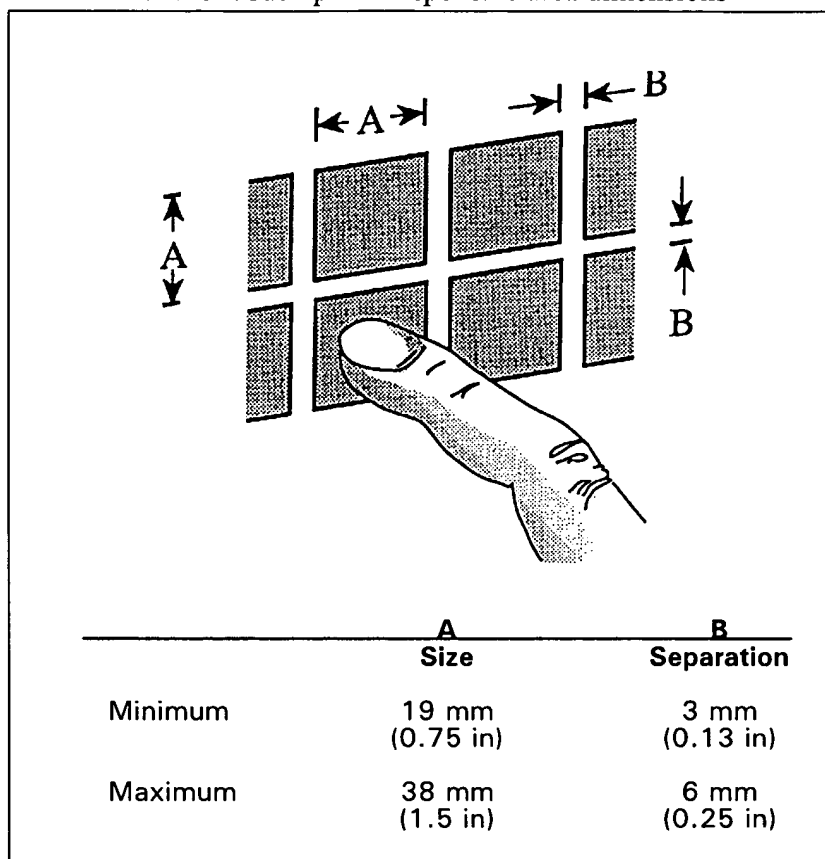
■ **8.8.4.2.7 Display feedback.** Display of user command or action feedback for touch panels shall not exceed 0.25 seconds.

8.8.4.3 Voice control

□ **8.8.4.3.1 Phonetically distinct vocabulary.** Spoken entries used for transactions should be phonetically distinct from one another. Testing should be performed to determine which sounds and words or phrases can be distinguished reliably.

> **Discussion.** Spoken command entries are not to be chosen arbitrarily. Tradeoffs between phonetic distinctiveness and familiarity of terminology need to be evaluated.

■ **8.8.4.3.2 Easy error correction.** Feedback and simple error correction procedures shall be provided for speech input so that if a spoken entry has not been correctly recognized by the computer, the user can easily cancel the entry and try again.

■ **8.8.4.3.3 Alternative entries.** Alternative input devices shall be available so that if the system cannot recognize a voice entry

after repeated attempts, another type of input entry can be substituted.

**8.8.5
Interchangeability
among input devices**

The interchangeability among input devices by the user can be useful during specific operations. Users may want to perform some actions using a keyboard and others actions using a pointing device. The ability to choose which input device must be optional to the user and not a requirement by the system.

□ **8.8.5.1 Redundant control.** If more than one input device is present, a user should be able to control computer interaction with all of them. For example, a keyboard should be capable of executing navigation and selection operations when used in conjunction with a mouse, light pen, or other input devices (see also 8.8.1.8).

> **Discussion.** Full interchangeability is not required. It is assumed that a user will select the input device that is most appropriate for the task being performed. For example, a user may rely on direct manipulation, using a pointing device such as a mouse or trackball, as the primary means of interaction for object selection and manipulation. Similarly, a user may use a keyboard primarily for text entry and for object selection being performed in conjunction with or interspersed with text entry.

**8.9
Accommodating
people with
disabilities**

The "Americans with Disabilities Act of 1990" (Public Law 101-336) prohibits employment discrimination against qualified individuals with disabilities. If a person's disability creates a barrier to employment, the Act requires that the employer consider whether reasonable accommodations could remove the barrier. The intent of the Act is to permit people with disabilities to compete with people without disabilities on the basis of the same performance standards and requirements once such accommodations have been made.

In general, there is no clear division between people with and without disabilities; rather any single ability tends to be distributed as a continuous function, and any individual may be at the high end of the distribution for some abilities and at the low end for others. Still, there is a large and growing number of people with disabilities or functional limitations. One estimate is that between ten and twenty percent of the United States population have significant disabilities. Indeed, almost everyone will experience functional limitations sufficient to make the operation of equipment or systems difficult if not impossible at some time during their lives.

Disabilities are not necessarily inborn or permanent. They may be temporary consequences of injury or illness, and they may be determined by the immediate environment. For example, a person might not be able to see a control or display because of darkness or might not be able to hear an auditory signal because of noise.

Definitions. An **impairment** is a loss or abnormality of physiological or anatomical structure or function. A disability is a physical or mental impairment that substantially limits one or more of a person's major life activities. A **person with a disability** is a person who has a disability, has a record of a disability, or is regarded as having a disability. A **qualified person with a disability** is a person who meets legitimate skill, experience, education, or other requirements of an employment position that he or she holds or seeks, and who can perform the essential functions of the position with a reasonable accommodation, if necessary. A **reasonable accommodation** is any modification or adjustment to a job or the work environment that will enable a qualified person with a disability to participate in the application process and to perform essential job functions. It may include: (1) making existing facilities readily accessible to and usable by people with disabilities, (2) restructuring jobs, (3) providing part-time or modified work schedules, (4) acquiring or modifying equipment or devices, (5) adjusting or modifying examinations, training materials, or policies, (6) providing qualified readers or interpreters, and (7) other similar accommodations.

In many cases, there are simple and low-cost (or even no cost) adaptations to equipment and systems that can significantly increase their accessibility and usefulness to people with disabilities. The most economical approach appears to be to design equipment and systems so that they are accessible to as many people as possible or practical. This is most easily accomplished when accessibility is considered during the design of the equipment or system.

A word of caution is in order: initial attempts at accessibility are sometimes made piecemeal; that is, features are made accessible rather than the equipment or system as a whole. For example, one feature might be made accessible to people with visual disabilities, and another feature to people with hearing disabilities, with the result that the equipment or system is not fully usable by either group. In most cases, it is possible, with careful design, to create equipment or systems that are simultaneously accessible to people with different types of disability. In any case, care must be taken to ensure that all the functions of the equipment or system are accessible to the desired populations of users.

Anything that is done to make equipment or systems more accessible is likely to be of benefit to all users, not just those with disabilities. For example, dips in curbs at pedestrian crossings of roadways were originally intended to accommodate people in wheelchairs, but they have also been of benefit to people with baby carriages, strollers, shopping carts, bicyclists, and pedestrians in general.

8.9.1 General

❑ **8.9.1.1 Equal access.** People with disabilities should have access to the same electronic office equipment, data bases, operating systems, and application programs as people without disabilities.

> **Discussion.** With respect to **input**, access might be achieved by providing:
>
> a.   alternative input mechanisms, including alternatives to simultaneous keystrokes, automatic repeat on depression of a key for a period of time, and alternatives to a mouse,
>
> b.   the capability of connecting an alternative input device, or
>
> c.   nonvisual keyboard orientation aids.
>
> With respect to **output**, access might be achieved by providing:
>
> a.   auditory alternatives to visual outputs,
>
> b.   visual alternatives to auditory outputs, or
>
> c.   access to the screen memory for the purpose of enlarging the display, converting text to speech, or converting graphics into an auditory representation.

❑ **8.9.1.2 Equal computing capability.** People with disabilities should have essentially the same computing capability as people without disabilities in the same position and office.

❑ **8.9.1.3 Support in manipulating data.** People with disabilities should be supported in manipulating data so as to attain end results equivalent to people without disabilities.

8.9.2 Accommodating people with moderate physical disabilities

Most of the difficulty experienced by people with physical disabilities in using computer systems stem from using input devices, such as a keyboard or a mouse, and from handling storage media, such as computer diskettes.

❑ **8.9.2.1 Multiple, simultaneous activations.** If a system requires multiple, simultaneous activations, such as the simultaneous depression of two or more keys on a keyboard, the system should provide an optional, alternative mode of operation.

> **Example.** One possible alternative mode of operation would accept sequential rather than simultaneous activations.

□ **8.9.2.2 Timed responses.** If a system requires a response in less than 5 sec or the release of a key in less than 1.5 sec, the system should provide either a means by which a user can adjust the time interval or an alternate mode that does not have the time requirements (see also paragraphs 7.6.3.5 and 8.9.2.8).

□ **8.9.2.3 "Pointing" from the keyboard.** A system that uses a pointing device, such as a mouse, should include a means for carrying out all of the pointing functions from the keyboard.

□ **8.9.2.4 Cursor control devices.** The select key on a cursor control device should have toggle capabilities, either as a standard feature or as a user-configured option, that allow a user to operate the device in a "button down" mode.

> **Discussion.** People with disabilities are likely to have difficulty simultaneously holding a select button down and moving the device, for example, in "dragging" an object in a graphical display.

□ **8.9.2.5 Minimal number of "small" targets.** The number of small targets should be minimized, especially if they are likely to be the objects of drag operations.

> **Discussion.** The difficulty of moving a pointer onto an object and of moving an object increases as the size of the object decreases, and the difficulty is greater for people with disabilities than for people without disabilities. If small objects cannot be avoided, a zooming capability might be provided (see also paragraph 8.9.4.1).

□ **8.9.2.6 Handling insertable and removable parts.** System components intended to be insertable and removable, such as computer diskettes, should require minimal reach and minimal manual dexterity.

□ **8.9.2.7 Controls and latches.** Controls and latches that are used regularly in the operation of a system should be accessible and operable with minimal reach and minimal manual dexterity.

□ **8.9.2.8 Avoiding inadvertent operation.** A computer or computer system intended to be operable by people with moderate motor disabilities should provide either a means for delaying the acceptance of a keystroke for a preset, adjustable amount of time (see also paragraphs 7.6.3.5 and 8.9.2.2) or a keyguard or means for mounting a keyguard.

> **Definition.** A **keyguard** is a keyboard cover with holes over keys the user is allowed to operate.

8.9.3 Accommodating people with severe physical disabilities

For people with severe physical disabilities, modifications to standard input devices may not be sufficient to allow them to use a computer. In these cases, a means for connecting an alternative keyboard, mouse, or other input device may be required.

❑ **8.9.3.1 Connection point for alternative input device.** A computer or computer system should provide a point at which an alternative input device can be connected. The computer should treat input from the alternative device the same as input from standard input devices.

> **Discussion.** One possible solution would be to provide a system command that would cause input from a standard serial, parallel, or other system port to be treated as if it had come from the computer's standard input devices.

**8.9.4
Accommodating
people with visual
disabilities**

Most of the difficulty people with visual disabilities have with computer systems arises in connection with output displays. Some difficulty also arises from input devices that require eye-hand coordination.

❑ **8.9.4.1 Enlarging a display.** People with visual disabilities should be provided a means for enlarging a display (see also paragraph 8.9.2.5).

> **Discussion.** This might be accomplished either by providing a means for attaching a larger display or by providing a means for enlarging all or part of the displayed image.

❑ **8.9.4.2 Selecting display colors.** If it is necessary to distinguish the color of graphics or text to understand displayed information, users should be able to select the colors used. However, see paragraph 8.2.4.1.17 on limiting user selection of colors.

❑ **8.9.4.3 Readability of lettering on keys and controls.** The lettering on keys and controls required for the operation of a computer or computer system should be large enough to be read easily and should have a distinct contrast with its background.

> **Discussion.** This might be accomplished by providing keycaps that can be removed easily and replaced with special keycaps for the visually impaired.

**8.9.5
Accommodating
people who are blind**

People who are blind usually have most of their difficulty with output displays. Some input devices also cause difficulty, for example, touch screens.

❑ **8.9.5.1 Connection point for alternative output devices.** Computers and computer systems should provide a point to which an alternative output device can be connected. Visually displayed information, both text and graphics, should be available at that point in a standard format.

> **Discussion.** The connection point might be a standard serial or parallel port. Alternative output devices include speech synthesizers and braille display devices.

□ **8.9.5.2 Alternatives to input devices.** If a computer or computer system has a standard input system that requires continuous visual feedback for operation, for example, a mouse or touch screen, the computer or system should provide an alternate means or mode for achieving as many of the input functions as possible. The alternative means or mode should be available at all times and should not require continous visual feedback.

> **Discussion.** It may not be possible to provide a reasonable alternative for some functions. For example, inputs such as free-hand sketching cannot be done easily without a device that requires eye-hand coordination.

□ **8.9.5.3 Nonvisual indication of state of toggle keys.** A computer or computer system should provide blind users with a nonvisual indication of the state of toggle keys. This indication may be available automatically or upon the user's request.

> **Discussion.** Probably the best solution from a blind person's point of view would be the use of switches that give a physical indication of their state, for example, toggle switches or rocker switches.

□ **8.9.5.4 Key demarcation.** All keys should have edges that can be discerned by touch. In particular, flat membrane keys without ridges outlining the keys should not be used.

□ **8.9.5.5 Identification of "home" keys.** The "home" keys of keyboards and keypads should have a distinct marking that can be discerned by touch.

□ **8.9.5.6 Key labels.** Optional or built-in nonvisual labelling of keys should be provided or available.

8.9.6 Accommodating people with hearing disabilities

People who have hearing disabilities and people who are deaf usually have little difficulty using computers. Most of the problems they do have can be eliminated by providing redundant visual outputs to tones and other auditory outputs.

8.9.6.1 General

□ **8.9.6.1.1 Redundant visual output.** All information required for system operation and error detection that is presented in auditory form should also be provided or available redundantly in an appropriate visual form.

□ **8.9.6.1.2 Hearing auditory outputs.** Computers and computer systems intended to be accessible to people with hearing disabilities should be designed to maximize the number of people who can hear auditory outputs (see also paragraph 7.6.2.1).

> **Discussion.** Auditory information (for example, synthesized speech, beeps, buzzers, tones, and machine

noises) may not be heard well enough to elicit the intended response. Possible solutions include:

a.   provide a volume adjustment,

b.   make auditory output as loud as practical,

c.   use sounds that have strong middle- and low-frequency components (500 - 3000 Hz),

d.   provide a headphone jack so that people with hearing disabilities can listen at high volume,

e.   provide a separate volume control for headphone jacks,

f.   place a sound source on the front of a device and away from sources of loud noise,

g.   include in the equipment a built-in inductive coil to facilitate the direct use of the telecoil in hearing aids,

h.   reduce the amount of nonmeaningful sound produced by the equipment, and

i.   present auditory information continuously or repetitively until the user responds to it.

8.9.6.2 Auditory screen representation

☐ **8.9.6.2.1 Granularity.** If a graphical interface is given an auditory representation, the auditory representation should be based on interface objects, not pixels.

☐ **8.9.6.2.2 Navigation.** Navigation in an auditory representation should move the user's position among different auditory interface objects.

> **Discussion.** Standard mouse movement is in terms of pixels, which have little or no meaning in an auditory representation.

■ **8.9.6.2.3 Hear and feel consistency.** The same type of object, such as a push button, shall have the same auditory representation and shall operate in the same way throughout an auditory interface.

☐ **8.9.6.2.4 Dual representation.** All interactions that a person without visual disabilities would see between the mouse cursor and objects on the screen should have auditory counterparts. These sounds may be simple or complex tones or patterns of tones, or speech.

❑ **8.9.6.2.5 Objects represented.** An interface given both visual and auditory representation should incorporate into the auditory representation at least the following objects if they appear in the corresponding visual interface:

a.   menus,

b.   windows,

c.   dialogs,

d.   buttons, and

e.   scroll bars.

❑ **8.9.6.2.6 Nonoverlapping objects.** An interface that is given both visual and auditory representation should not have objects that completely obscure other objects, for example, a window that completely overlaps another window.

❑ **8.9.6.2.7 Eliciting an object's name.** A user should be able to elicit the name of the object currently being pointed at.

> **Example.** Pressing one of the buttons of a mouse might result in a synthesized speech announcement of the name of the object.

❑ **8.9.6.2.8 Size and location of objects.** In general, users should not be able to move or change the size of objects in auditory representations.

8.9.7 Accommodating people who have seizure disorders

❑ **8.9.7.1 Avoiding flashing-induced seizures.** Computers and computer systems should maximize the number of people who can view an output display without experiencing a seizure (see also paragraph 7.6.2.8).

> **Discussion.** People who are sensitive to seizures may have seizures induced by flashing screen cursors or by flickering displays. The solution is to ensure that the flash or display refresh rate is as far above or below 15-20 Hz as possible or practical.

8.9.8 Accommodating assistive devices

❑ **8.9.8.1 Electronic documentation.** Manuals and other important documentation intended to be accessible to people with disabilities should be available in electronic as well as printed

form. This would permit presentation of the material on an assistive device such as an enlarged display, a speech synthesizer, or a braille reader. Both text and graphic information should be included (same as paragraph 10.6.1).

□ **8.9.8.2 Speech output compatibility.** Computers and computer systems should provide a built-in speech output capability or provide a point to which a speech synthesizer can be connected.

□ **8.9.8.3 Special display window.** A windowing environment should provide the capability of opening and maintaining a special window that can remain fully visible. Once displayed, the special window would be available continuously for use by special input routines.

□ **8.9.8.4 Connection point for switches.** Computers and computing systems should provide a point at which at least two momentary contact single-pole single-throw input switches can be connected.

> **Discussion.** Some alternative input devices require the connection of special switches or interfaces. Examples are "sip and puff" switches and eye blink switches. Unused pins on existing connectors could serve as these connection points. It would be desirable to be able to connect analog transducers as well as binary switches.

□ **8.9.8.5 Distinguishing macro input from typed input.** Computers and computing systems should be able to distinguish between typed, auto-repeat, and macro-generated "keystrokes" so that, if appropriate, they can be treated differently by the operating systems and application software.

> **Discussion.** "Keystrokes" generated by assistive devices or assistive software may be sent faster than the application software can recognize them, in which case, they may be ignored, thus preventing use of the assistive device or software.

□ **8.9.8.6 Keyguards.** Keyboards should be designed so that keyguards can be mounted easily.

> **Discussion.** It is desirable that the manufacturer of the keyboard also supply a compatible keyguard.

EVIDENCE APPENDIX J
COPY OF BRENNER ET AL., U.S. PATENT NO. 6,004,211